Identification of highly altered genes in Breast Cancer patients through the integration of multiple omics data

Rafael Riudavets Puig
Master in Bioinformatics and Biostatistics

Guillem Ylla Bou
David Merino Arranz

06/2018

GNU Free Documentation License (GNU FDL)

FICHA DEL TRABAJO FINAL

- **Título del trabajo:** Identification of highly altered genes in Breast Cancer patients through the integration of multiple omics data
- **Nombre del autor:** Rafael Riudavets Puig
- **Nombre del consultor/a:** Guillem Ylla Bou
- **Nombre del PRA:** Carles Ventura Royo
- **Fecha de entrega (mm/aaaa):** 06/2018
- **Titulación:** Màster en Bioinformática y Bioestadística
- **Área del Trabajo Final:** Integración de datos ómicos
- **Idioma del trabajo:** Inglés
- **Palabras clave:** omics, integration

Abstract

Biological data integration is a field that is gaining relevance in studies based on high-throughput technologies. This type of technologies let us explore an incredibly big amount of variables in our samples from different perspectives such as transcription of different genes, methylation events along the genome, copy number aberrations or mutations in the studied genome. The integration of the different perspectives might allow us to gain further insights into the states of a system compared to if we only used one single data type. In this work we have studied RNA-seq, Bi-seq, DNA-seq and SNP Array data from The Cancer Genome Atlas (TCGA) for 35 Breast Cancer patients that had samples for both tumor and normal tissues with the aim of exploring the effect of such an integration. We have further visualized the results by overlapping them with signaling networks that are known to be involved in cancer. On one hand, we have observed that some genes like PI3K, Raf, Wnt or FGF, which are known to be important in cancer progression, show alterations in several data types. On the other hand, we have observed that some genes that are also known to be very relevant in cancer show alterations only in one data type. An example of the latter would be P53, which only seemed to show mutations in the DNA-binding domain, but did not show alterations in the other data types.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

The advent of high-throughput (HT) technologies has revolutionized research in many areas, opening the possibility to obtain massive amounts of biological data with a minimum technological and economical effort. With these new possibilities there has been a huge effort from a lot of different fields to improve not only the experimental procedure itself, but also the statistical methodology used for the analysis of this type of data. A milestone in this revolution was accomplished with the development of microarray based technologies in the late 90's and early 2000's (Schena et al. 1995);(Bumgarner 2013). During the last decades, these methodologies have been experiencing a lot of progress, leading to an important development in functional genomics studies based on HT technologies (Alizadeh et al. 2000), (Vijver et al. 2002).

One of the most important characteristics of HT experiments is that they allow the identification and evaluation of a very big amount of variables in a single experiment. This property is both an advantage and a disadvantage, since one of the main challenges in the analysis of these types of data is the problem known as $n << p$, or the fact that we usually have many more variables than samples (Johnstone 2009).

In Molecular Biology, omics data are known as the result of applying a HT technology to study a specific type of biological molecular entity such as DNA, RNA, proteins, or others. Hence, an option to distinguish different types of omics data is to distinguish them by what kind of molecule we are studying. For example, the study of RNA molecules, or transcripts, based on HT technologies can be known as transcriptomics. We can also find higher classifications that involve several subtypes of omics data, such as genomics, proteomics or metabolomics among others. For instance, genomics can be understood as the study of genomics data, namely DNA sequence. Depending on what part of the genome we are focusing on, we will be in one subtype or another.

The basic workflow used in microarrays can be seen in **figure 1.1**. In it, we see how the molecule of interest is extracted from samples. Next, the molecule is transformed, amplified and labeled with fluorophores, followed by hybridizing the samples onto the microarray. The last steps will be to scan the microarray and analyze the results. For example, in **figure 1.1** the RNA molecules of a sample are isolated and reverse transcription is applied to obtain cDNA. The cDNA is then transcribed again, and the cRNA is labeled and fragmented, followed by hybrydization with the chip. The last step will be to scan the microarray to get the signal for the different fragments and analyze it. There are several microarray designs and manufacturers, Affymetrix being one of the most popular microarray manufacturers.

A research subject where microarrays have become very popular is cancer research. However, due to the elevated cost of this type of experiments, specially during the first years after after these technologies were developed, a lot of these studies were based on a single type of omic and did not have too many replicates. An example of these research lines is the use of transcriptomics data to identify potential biomarkers (Alizadeh et al. 2000); (Allison et al. 2006); (Dudoit and Fridlyand 2002); (Tusher, Tibshirani, and Chu 2001); (Barillot et al. 2012); (Golub et al. 1999); (Ramaswamy et al. 2003); (Veer et al. 2002); (Y. Wang et al. 2005), which is a biological molecule that can be objectively measured and allows us to identify an ongoing biological process such as a pathology, response to a treatment, etc. However, it is known that a lot of heterogeneity between tissues and

patients can be found (Ein-Dor et al. 2005); (Symmans et al. 1995); (Tomlins et al. 2005); (Mootha et al. 2003). An example of this is that we would expect to find differences in the transcriptomic profiles between two samples derived from different tissues. This type of variability is also known as Biological Variability.

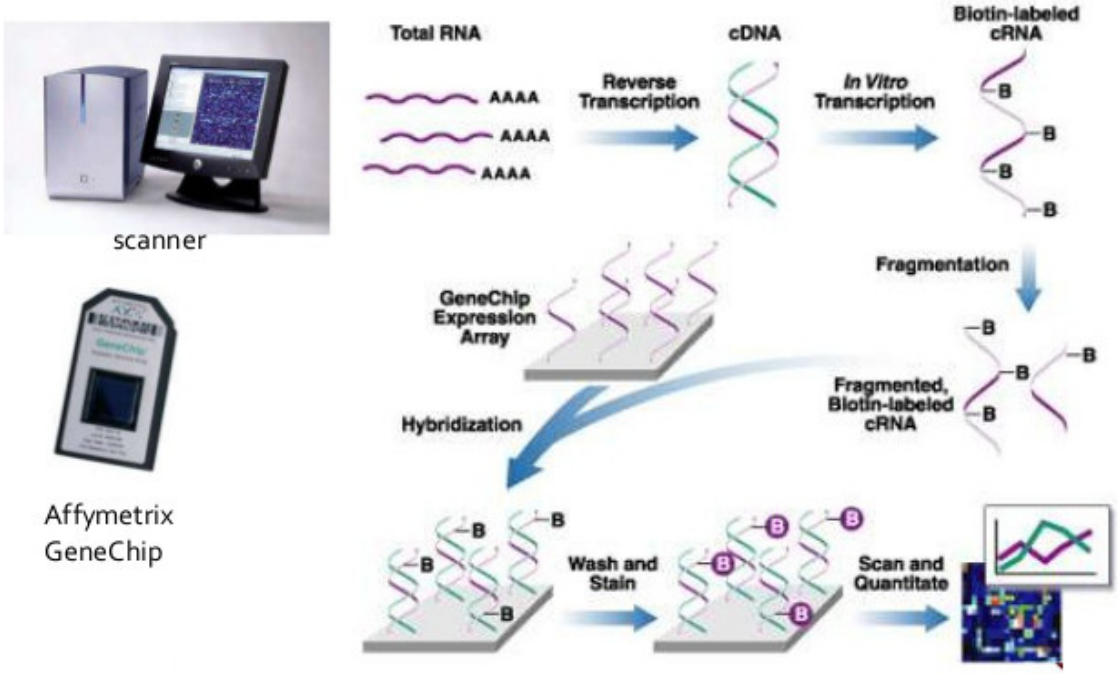


Figure 1.1: Main steps on a microarray based experiment. Image extracted from http://slideplayer.com/slide/8024786/. Briefly, the molecule of interest is purified, fragmented, labeled and put in contact with the probes, who will specifically bind to to the fragments. Next, signal will be detected and quantified for further study.

Although microarrays are still popular, they are starting to be replaced by a new type of HT technology known as Next Generation Sequencing (NGS) (Ledford 2008). This type of methodology allows the sequencing of very high amounts of nucleotide sequences in a short time and relatively low cost. Some examples of this type of technology are DNA-seq, which allows the study of the sequence of DNA molecules, or RNA-seq, which allows the sequencing of RNA molecules in a sample. The main process in NGS (**Figure 1.2**) is to fragment the molecule that we want to study, followed by sequencing each fragment, and reassembly of the fragments or the alignment of the fragments against a reference sequence. If applied to RNA, for instance, we would first purify the RNA in our sample, followed by fragmenting it, sequencing it, and then realigning the sequenced fragments with a reference genome to extrapolate the expression of each gene based on the number of RNA-seq reads mapped to it. After the necessary statistical processes to make the samples comparable, named normalization, we could perform a comparison between conditions, which would let us study the differential expression of genes across conditions.

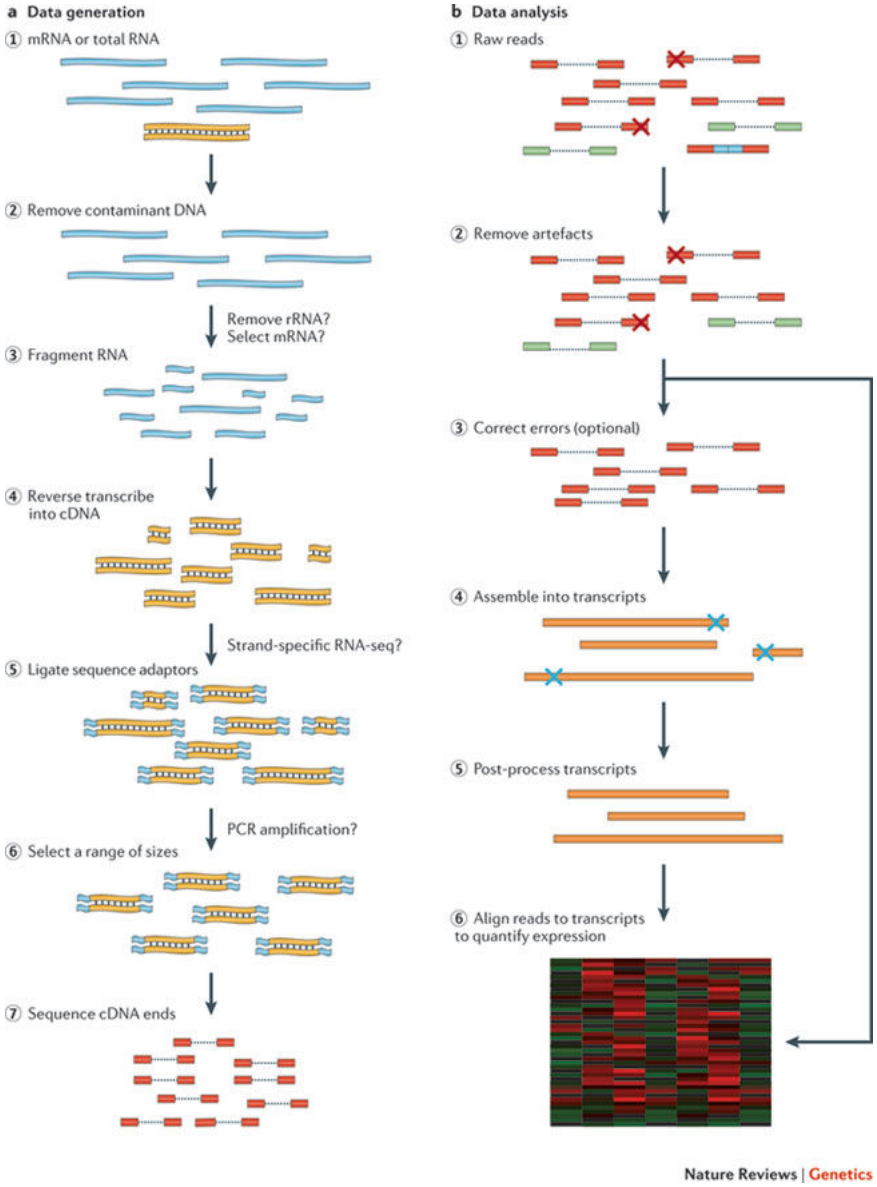


Figure 1.2: Main steps on an RNA-seq based experiment, extracted from (Jeffrey and J. Wang 2011). Briefly, these steps are purification of the RNA molecules, fragmentation and sequencing of the fragments. Next, the sequenced fragments are aligned to a genome to locate and quantify the sequenced fragments.

Due to reproducibility reasons as well as to the cost of HT technologies there has been an effort to make the data of these experiments available to the scientific community. This has led to the development of repositories specialized in the storage of HT experiments data. Some examples of these repositories are Gene Expression Omnibus (Edgar, Domrachev, and Lash 2002), The Cancer Genome Atlas (Weinstein et al. 2013), Array Express (Kolesnikov et al. 2015), International Cancer Genome Consortium (Hudson 2010), Cancer Cell Line Encyclopedia (Barretina et al. 2012).

A challenge for research based on HT technologies, either based on microarray or NGS data, is the statistical analysis of the data. A lot of effort has been made to both understand the nature of this type of data and develop better strategies to analyze and interpret it. A widely used tool in the analysis of these types of data is R (R Core Team 2018), an open-source statistical software based on a command-line interface. A proof of the effort that was developed in this direction is the creation of the Bioconductor project (RC Gentleman et al. 2004), which is an R based project whose

objective is to provide tools for the analysis of high-throughput data in the field of Biology. Some tools that have been developed and that got very popular are limma (Ritchie et al. 2015), DESeq2 (Love, Huber, and Anders 2014), edgeR (Robinson, McCarthy, and Smyth 2010), Minfi (Aryee et al. 2014), and many others. There are other initiatives that have the same objective, such as Galaxy (Goecks et al. 2010). An advantage for the latter is that it provides a graphical interface, breaking the boundary of a command-line based interface that stops some users from getting into the use of R.

A branch of biology that has experienced an important development thanks to the development of the aforementioned technologies is Systems Biology. The main objective of this field is to understand biology at the system level (Kitano 2002). From a Systems Biology point of view, we might see each omics data type as an incomplete image of the state of the system that we want to study (Pavel, Sonkin, and Reddy 2016). This has been an incentive to develop new strategies that try to complement the "incompleteness" of each data type with other data, a process known as data integration and whose objectives are to gain even further insights into the biology behind our data. Some examples of this are the creation of studies where we look for differentially expressed genes, followed by merging the results with a network of protein-protein interaction networks (PPI). In this network, each node represents a protein and each edge represents an interaction between the proteins it is linking. This would allow us to see what genes are differentially expressed, the direction (upregulation/downregulation), and the proteomic context around these genes. We could afterwards try to find groups of interconnected nodes, or modules, in our network that show a differential expression. Other strategies have gone even further, integrating several types of omics data with network topologies. An example of that is PARADIGM (Vaske et al. 2010), which takes Copy Number Alteration and Transcription data together with a network topology to infer the activity state of known pathways through a probabilistic graphical model. Another strategy is ROMA (Martignetti et al. 2016), which uses transcriptomics and gene set data to infer the activity state of a given gene set through the statistical assessment of overdispersion and overcoordination of the transcription of the genes in the gene set. A third example would be VIPER (Alvarez et al. 2016), which uses transcriptomic data to infer the activity state of specific proteins. A final example is FEM (Jiao, Widschwendter, and Teschendorff 2014), which computes the differential expression and methylation across two conditions, followed by integrating it with a PPI network. During this process of integration, a scoring is assigned to each node depending on the results of the statistical analysis. Then, a score will be computed for each edge connecting two nodes as a function of the score of the nodes it is connecting. The final step is to find groups of nodes and edges that display a higher score than the average of the whole network to detect what they called Functionally Enriched Modules.

Although it is common to use transcriptomics data as a proxy of protein activity levels, it is not completely clear how much of the activity of a protein can be predicted only from the transcription of the gene encoding for it. A reason for that is the huge amount of regulatory events that can happen between transcription and protein activation, such as post-transcriptional regulation (miRNAs, for instance) or post-translational modifications (phosphorylation, acethylation, etc) that regulate the activity of a protein. However, some studies use the transcription of genes as a proxy of the activity of the Transcription Factor regulating them (Garcia-Alonso et al. 2018). Other researchers also incorporate the use of time-series data to consider the dynamics state changes of the system (Wachter and Beissbarth 2016).

It is also important to acknowledge that although we are studying these types of data independently, they are all interconnected. For example, if we had a region of the genome that had suffered a deletion, we would expect to observe a decrease in the transcription of the genes in the affected region. If the aim of a study was to find potential causal reasons for the low expression of those genes and we only had transcriptomics data, we would not be able to reach such an explanation based on the data.

Chapter 2

Motivation and Objectives

I have several motivations to want to develop this Master Thesis. First of all, I believe in tailored medicine and its potential. The finding of pathways or proteins who look to be the most involved in the cause of a disease could be used as an indication of the most appropriate treatment for each patient, leading to maybe an increase of the efficiency of treatment assignment processes and thus having an impact on society's health. On the other hand, HT technologies are proving to be a very powerful tool that is making them a key element to advance in understanding the mechanisms of complex diseases. However, although their potential, they still have their limitations. The integration of several omic data types might allow us to develop more complex hypotheses and get more insights into complex diseases, allowing researchers to advance in the field.

Our hypothesis in this work is that the integration of different omics data will allow us to identify genes that show alterations at various levels that might affect their function. The considered levels will be transcriptional changes, methylation events in their promoter, copy number variations and mutations. In order to do so, we will use RNA-seq, Bi-seq, SNP Array and DNA-seq data from a subset of Breast Cancer patients from TCGA. We will perform individual analyses on each type of data and then integrate the results, followed by overlapping them with some networks known to be involved in cancer.

Taking all this into account, the objective of this thesis is to identify genes that show many types of alterations and assess their relevance for pathways related to cancer. This will be done through the following steps:

- Find differentially expressed genes between tumor and normal conditions.
- Find differentially methylated probes that map within 200 bp of the Transcription Start Site of a protein coding gene.
- Find genes affected by recurrent copy number variations in the tumor samples.
- Find genes showing mutations that could affect their function.
- Get a score for each data type for each single gene.
- Integrate the scores for a gene to get a single score per gene.
- Visualize the results in some networks known to be involved in cancer.

Chapter 3

Material and Methods

3.1 Statistical software

Most of the data analyses from this thesis were done with R (R Core Team 2018)(version 3.5.0). Within R, several different packages have been used for each part. The list of packages, its references and their application is as follows:

- TCGAbiolinks (Colaprico et al. 2015): used to query the TCGA repository, download its data and perform the Differential Expression Analysis and Differential Methylation Analysis.
- maftools (Mayakonda and Koeffler 2016): used to manipulate the maf files, analyze and visualize them.
- Gaia (Morganella 2010): used to study the recurrence of the copy number variations across the tumor samples.
- GenomicRanges (Lawrence et al. 2013): used in the process of annotating the results of the recurrent Copy Number Variations.
- stringr (Wickham 2018): used in the preprocessing of some files to give them a format that was easier to work with.
- org.Hs.eg.db (Carlson 2018): database used to find annotations for genes.
- GOstats (Falcon and Gentleman 2007): to perform Gene Ontology terms enrichment analysis.
- biomaRt (Durinck et al. 2005): to get annotations in different parts of the process.
- Pathview (Luo et al. 2013): to produce images with KEGG pathways and the altered genes in them.

3.2 Data query and download

The data used for this study has been downloaded from TCGA using the TCGAbiolinks package. In the following sections we will describe each data type and its format.

TCGA provides data in different levels of processing, being some of those levels not of public access. Although some of those levels have restruicted acces, all the data used in this work is publically available. The information about each data type, its format and its processing pipelines can be found in the GDC Data User's Guide, at https://docs.gdc.cancer.gov/Data/PDF/Data_UG.pdf.

We selected Breast Cancer samples from patients that had data for Transcription, Methylation, Copy Number Variation and Mutations in both Tumor and Normal tissues. This left us with an initial set of 37 patients. After exploring the data, we decided to discard some of the samples due to some pointers indicating these might be outliers. Hence, the final number of samples was 35. The code to find and download these samples can be found in appendix A.1 and appendix A.2. The code to explore all the data types and update the samples we will use is found in appendix A.3.

3.2.1 RNA-seq

Downloaded data

HT-Seq counts (Simon Anders, Pyl, and Wolfgang Huber 2014) for normal and tumor samples where downloaded for the patients described above.

Data exploration

First, histograms of raw counts and log2 of the raw counts were plotted to get a general idea of the distribution of the values. Since some genes could show zero counts, a pseudocount was added to avoid problems when calculating the log2. Second, Principal Component Analysis was run on the data to check for potential outliers and possible clusters.

Analysis

We used the TCGAbiolinks package to perform the preprocessing, normalization, filtering and differential expression analysis (for more details, see below). Next, we performed a Gene Ontology term enrichment analysis with the package GOstats. Finally, we used the pathview package to visualize the results for the previously mentioned KEGG pathways.

HT-Seq counts preprocessing was done by applying an Array-Array Intensity Correlation with a pearson correlation threshold of 0.6. Samples showing a sample-sample correlation below the given threshold might be understood as possible outliers. Next, between and within lane normalizations were applied using the normalization function from TCGAbiolinks, which calls the EDASeq package (Risso et al. 2011). The next step was to apply some filtering. This was done by removing all transcripts showing a mean expression across all samples below the lower quartile of the mean across all samples.

Differential Expression Analaysis was performed using a FDR threshold of 0.001 and a logFC threshold of 1. The method used was glmLRT, which calls the edgeR package to perform the differential expression analysis. This analysis is based on fitting a negative binomial generalized log-linear model to the read counts for each gene.

The Functional analysis was performed by using the GOstats R package. We looked for enriched Gene Ontology Biological Process Terms in the differentially expressed genes. The selected p-value threshold for the enrichment analysis was of 0.001.

Finally, the pathview package was used to get the visualizations of the results. The logFC of the differentially expressed genes was used to provide a colour scale for the image, where green nodes represent negative logFC values (underexpression) and red nodes represent positive logFC values (overexpression).

The script to reproduce the complete analysis of the RNA-seq data can be found in the appendix A.4.

3.2.2 Methylation

Downloaded data

β-values were downloaded for normal and tumor samples for the patients described above. The platform used in this study is the Illumina Human Methylation 450k, which is based on bisulfite sequencing. β-values are calculated from array intensities by applying $\beta = M/(M + U)$, where M is the intensity of the probe assessing a methylated CpG, and U is the intensity of the probe measuring the unmethylated CpG.

Data exploration

The exploration of the methylation data consisted of a boxplot representing the mean methylation value for the normal and tumor conditions, a Principal Component Analysis and a plot with the density of the β-values for 10000 randomly picked probes of each sample. To ensure complete reproducibility of the figures, the set.seed(444) R command was used to pick the samples.

Analysis

The Differential Methylation Analysis was performed using the TCGAbiolinks package with a p-value cut of 0.001 and a minimal difference in the mean value of a probe between conditions of 0.35. From the resulting differentially methylated probes, we selected the closest probe mapping within 200 bp of a TSS of protein coding genes. This allowed us to end up with a single value per gene.

Next, we performed a Gene Ontology term (Biological Process) enrichment analysis to see if there were some functions that seemed to be enriched in our list of differentially methylated genes. The selected significance threshold to detect a term as enriched was 0.001.

We also visualized the results using the pathview package to get an image as the one described in the previous section. In this case, the colour scale was converted to a 1/-1 scale, where 1 (red) represented hypomethylation of the probe annotated to the gene, and -1 (green) represented hypermethylation of the probe annotated to the gene.

The script to perform the whole analysis of the methylation data can be found in the appendix A.5.

3.2.3 Copy Number Variation

Downloaded data

Segment Mean Values for the tumor samples were downloaded for the selected patients. Segment values are calculated with $Segment\ mean = log_2(copy\ number/2)$. This ensures that diploid regions will have a value of 0, and amplified or deleted regions will show positive or negative values, respectively.

Data exploration

In this case, the exploration consisted in plotting the density of the segment mean values for normal and tumor samples to get an idea of the distribution of the values across the samples.

Analysis

Recurrent CNV analysis was performed using Gaia (Morganella, Pagnotta, and Ceccarelli 2011), which applies a permutation test to assess the significance of the recurrence of a specific CNV, together with considering within-sample homogeneity to identify the most significant peaks. To perform the analysis, the SNP6 GRCh38 Liftover Probeset File for Copy Number Variation Analysis was downloaded from the GDC website (https://gdc.cancer.gov/about-data/data-harmonization-and-generation/gdc-reference-files) on the 25th of April of 2018.

Gaia requires data labeled with amplified/deleted regions as input. Hence, regions showing a segment mean higher than 0.4 were labeled as amplified regions, while regions showing a segment mean lower than - 0.4 were labeled as deleted regions. To run Gaia, a q-value threshold of 0.0001 was selected together with an h-value threshold of 0.12 and 1000000 iterations. Since this is based on a permutation test, we set a seed of 444 using the set.seed(444) R command to ensure full reproducibility of the results.

The annotation of the genes in the affected regions was done with the TCGAbiolinks and GenomicRanges packages by querying Biomart (hg38 assembly).

The GOstats R package was used to perform a functional analysis of the recurrently amplified/deleted genes, for which we looked for enrichment of Gene Ontology Biological Process terms using a p-value threshold of 0.001. We also used the pathview package to display the recurrently altered genes in the described pathways. We again transformed the results to a numeric scale, where 1 would represent recurrently amplified genes (red), and -1 would represent recurrently deleted genes (green).

The script used to perform the whole analysis can be found in the appendix A.6.

3.2.4 DNA-seq

Downloaded data

Simple Nucleotide Variation data detected by using the MuTect2 (Cibulskis et al. 2013) was downloaded from TCGA. This data has been obtained from a Whole Exome Sequencing (WXS) analysis, which means the sequencing has been performed only in the exomic sequences of DNA. The workflow is designed to identify somatic variants through the comparison of the frequency of each allele in normal and tumor samples. Next, each mutation will be annotated, and several cases will be aggregated into one single MAF file. The downloaded MAF file contained the information for many more patients than the ones we wanted to use in our study. To subset the data, we used the subsetMaf function from the maftools package, obtaining the mutation information only for the patients we wanted.

Data exploration

The exploration of the data was based on a description of the amount of different mutation types, namely frame shift deletions/insertions, in frame deletions/insertions, missense mutations, nonsense mutations, nonstop mutations and translation start site mutations. We also described the impact that the different mutations were annotated to have on the function of the protein encoded by the gene. Furthermore, we used the maftools package to plot a summary of the different mutation types, and most frequently mutated genes. Finally, we created a lollipop plot for the two most frequently altered genes, which shows the amino acid sequence of the corresponding proteins and its different domains together with the points where the mutations were detected in our samples.

Analysis

We wanted to perform a significantly mutated genes analysis with the mutations data using MuSiC (Dees et al. 2012). However, this software needs the BAM files created during the analysis of the DNA-seq data. Although these files are stored in the TCGA Data Portal, they are not of public access. It is for this reason that we decided that the analysis of this data type would be much more descriptive. This will also be considered in the integration part, where the weight assigned to the score of the DNA-seq data will be lower than the weight of the rest of the scores.

We calculated a score for each gene based on the proportion of samples showing a mutation on each gene. The formula used to calculate the score is the following:

$$score = 1/log_{10}(\frac{number\ of\ mutated\ samples\ for\ gene\ i}{total\ number\ of\ samples})$$

This would give us a score with values ranging from 0 to infinity, where higher values would be representing genes that were mutated in a higher proportion of samples. This score was then used with the pathway package to create a map of KEGG pathways and the scores for the genes in them.

The code for the exploration and analysis of this data can be found in appendix A.7.

3.3 KEGG Pathways

In order to integrate the results of the previous omics analysis within the KEGG pathways, we used the R package "pathview".

We selected 5 pathways that are known to be involved in Cancer. These are:

- Cell Cycle (KEGG ID: hsa04110): a network containing the most important genes involved in the mitotic cell cycle.
- P53 signaling pathway (KEGG ID: hsa04115): a signaling pathway involved in the detection of DNA damage. When DNA damage is detected, the cell can promote the arrest of the cell cycle to try to repair the damage in the DNA. However, if this damage is not repairable, the cell will be sent to apoptosis.

- PI3K signaling pathway (KEGG ID: hsa04151): a pathway known to promote cell cycle progression through the reception of growth factor by Tyrosine Kinase Receptors (RTKs).
- MAPK signaling pathway (KEGG ID: hsa04010): a pathway known to be involved in cell proliferation and migration.
- Breast Cancer network (KEGG ID: hsa05224): KEGG has a network of the most important networks described in Breast cancer. These networks are separated to illustrate the Luminal B, HER2 positive and Triple Negative Breast Cancer subtypes. This map will be used in the integrated analysis, since it contains a summary of the most important parts of some of the pathways above.

3.4 Integration of the data

The integration of the data was performed by computing a single score for each significant gene in each data type. These scores were then summed to obtain a single score for each gene. The summation of scores was weighted to account for the fact that the DNA-seq data was only descriptive. Thus, transcription, methylation and copy number variation data had all the same weight (0.3), while the mutation data had a lower weight (0.1) on the final score. The values used to give a score to each omics data type were: logFC for transcriptomics, methylation state for epigenomics, aberration type for CNVs, and proportion of samples showing a mutation for DNA-seq. The final score would be an indicator of the accumulation of aberrations in the different genes in the pathways studied. The values for this score would range from 0 to 1, where 0 would mean that no aberrations were detected in any omics data type, and 1 that aberrations had been found in all data types.

$$Final\ score_i = 0.3(Score_{Trans_i} + Score_{Meth_i} + Score_{CNV_i}) + 0.1Score_{Mut_i}$$

We re-scaled the different scores so they were all in the same range of values, since this was necessary to make sure that no omics data type had a higher weight than the others only due to the range of its scores. The final score would then be an indication of the amount of aberrations observed in each gene, without considering direction.

Epigenomics data was processed so 1 would indicate that a differential methylation state had been observed between conditions and 0 would mean that no change had been detected.

A very similar approach was followed for the copy number variation data, where 1 would mean that recurrent amplifications/deletions had been detected and 0 would mean that no aberrations were detected for that gene.

In the case of mutation data, the score was computed as specified in section 4.3.1. Next, the score was re-scaled to a 0 to 1 scale by applying min-max scaling, which is done by

$$x' = \frac{x - x_{min}}{x_{max} - x_{min}}$$

.

Finally, the absolute value of the logFC of the transcription data was re-scaled to a 0 to 1 scale by applying again a min-max scaling.

The pathview package needed a higher range of values for the different scores. Thus, we transformed the scores through:

$$Final\ score\ trans_i = |1/log_{10}(Final\ score_i)|$$

This resulted in a final range of scores that went from 0 to the maximum score. Once we had a score per gene, we plotted the studied pathways, where a lower score was represented in gray, indicating a lower amount of alterations in that gene, and a higher score was represented in red, indicating a higher amount of observed alterations in that gene.

3.5 Workflow summary

A figure representing the whole workflow can be found in **Figure 3.1**. In it, we can see the main steps that are the base of this thesis. First, we find TCGA Breast Cancer samples that have all the data types for both normal and tumor samples. The next step is to download all the data for those samples. Following, we will start analyzing each data type individually. The steps in this analysis for each data type are a first exploration of the data, preprocessing and normalization if needed, and find the relevant genes in each data type. Depending on the data type, we will perform different analyses:

- DNA-seq: we will compute the proportion of samples showing a mutation in each gene.
- RNA-seq: we will perform a Differential Expression Analysis to find genes that are significantly over/underexpressed in the tumor samples.
- Bi-seq: we will perform a Differential Methylation analysis for each probe.
- SNP Array: we will look for recurrent Copy Number Variations in our tumor samples.

Once we have the results of the analyses, we will visualize them with the pathview package to see if the nodes in the networks of interest show alterations on the different omics data types.

Finally, we will integrate the results by computing a score for each gene that considers the results of the different data types. This score will let us see if a gene is altered in a lot of omics data types or just in one or more. The score will be used to visualize the same networks, with a colour scale indicating the amount of observed aberrations in each gene of the pathway.

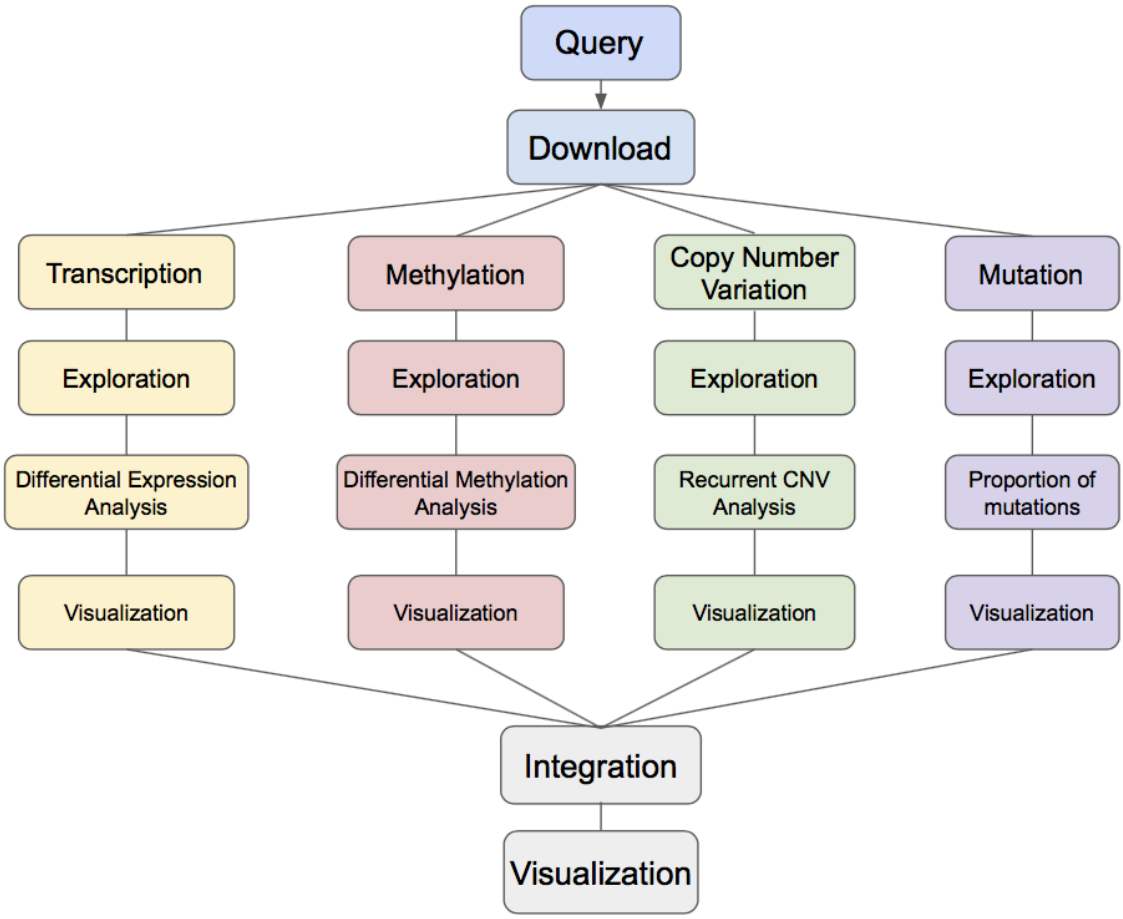


Figure 3.1: Summary of the pipeline followed during the work.

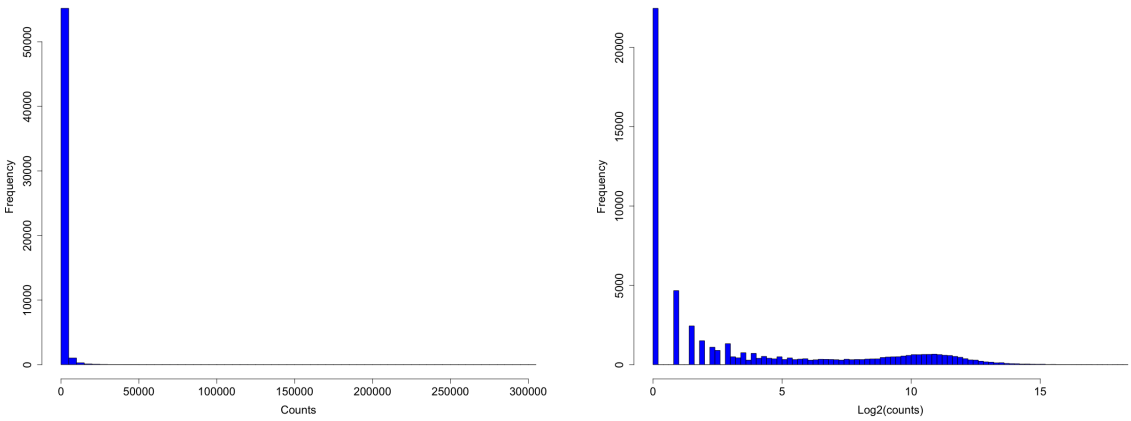
Chapter 4

Results

4.1 RNA-seq

4.1.1 Exploratory Analysis

We started by plotting histograms of both raw counts (**Figure 4.1a**) and log2 transformed counts **(Figure 4.1b)** in order to get an idea of the distribution of the values.



(a) Histogram of the transcription raw counts

(b) Histogram of the log2(counts)

Figure 4.1: Histograms of raw (a) and log2 (b) raw counts

Due to the presence of very high count values, the histogram in **figure 4.1a** is not very informative. However, in **figure 4.1b** we see that a log 2 transformation allows us to gain more insights into the distribution of the counts. We also observe that the biggest amount of genes show low values of transcription, while some others are showing much higher values in their expression.

Next thing we wanted to see is if there would be any distinguishable pattern appearing if we performed a Principal Component Analysis on the data. Thus, we applied PCA on the raw counts and coloured differently Normal and Tumor samples. The results of plotting the PC1 vs. the PC2 can be seen in **figure 4.2**. In it, we see that some samples cluster a bit far away from the main mass of points. Thus, we identified these samples and studied them more deeply. After further checking them, we saw that they had all been analyzed at the same center, and that they where the only samples analyzed by that particular center. An explanation of these results would be that there had been some differences in the manipulation or during the analysis of the samples. Since we saw those samples plotting far from the rest, and we had seen that they were the only ones having been analyzed at that specific center, we finally decided to discard them. The reason for this decision is

that we could not know if the differences we saw where due to biological or technical reasons.

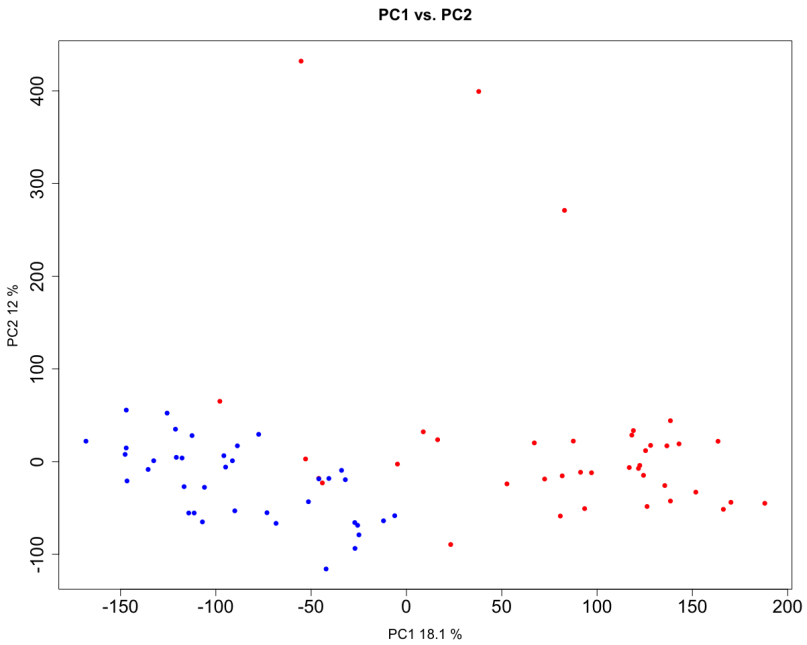


Figure 4.2: PC1 vs. PC2 of the Principal Component Analysis of the transcription data: red dots represent tumor samples, while blue dots represent normal samples.

A second PCA, now without the discarded samples, can be seen in **figure 4.3**. In it, we can see that the data seems to be much more uniform now. We can also see that the PC1 separates the normal (blue dots) and tumor (red dots) samples fairly good. An interpretation for that could be that the transcriptomic profiles of normal and tumor samples are different, which is what we would expect.

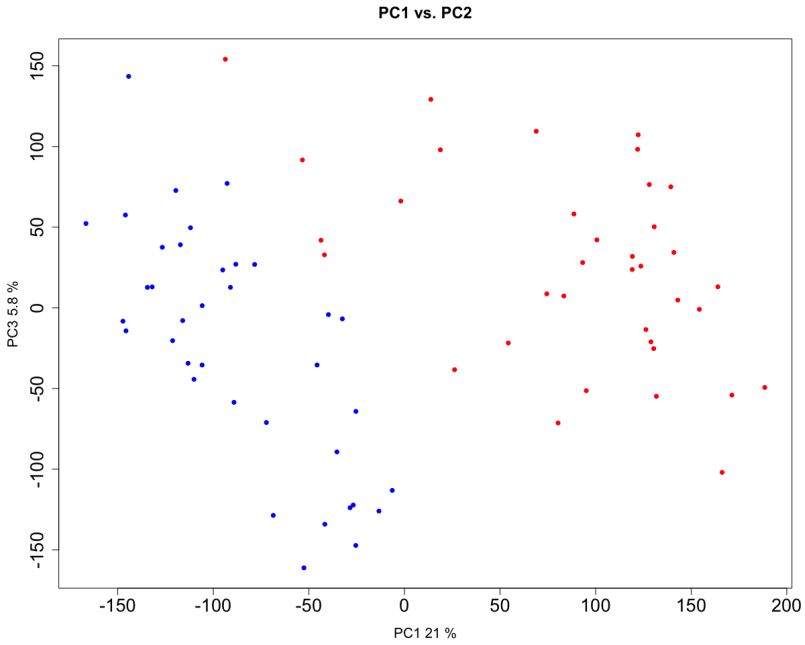


Figure 4.3: PC1 vs. PC2 of the Principal Component Analysis of the transcription data after removing the potential outliers: red dots represent tumor samples, while blue dots represent normal samples.

4.1.2 Differential Expression Analysis

The differential expression analysis begun with preprocessing, normalizing and filtering the data. As mentioned in Material and Methods chapter, the preprocessing was performed by using an Array-Array Intensity Correlation. The results of the preprocessing can be seen in the **appendix figure B.1**. We saw that the lowest sample-sample correlation was 0.84, which was interpreted as a sign of no more outliers in our data.

Following, within and between lane normalization was applied. The boxplots of before (top) and after (bottom) can be seen in the **appendix figure B.2**. These results let us see that the normalization procedure has been realized successfully.

Finally, the last preprocessing step was to apply quantile filtering, where genes showing a mean expression across samples below the lower quartile of the mean across all samples would be filtered out. The next step was to perform a Differential Expression Analysis. We ended up with a total of 4130 differentially expressed genes, for which we performed a functional analysis to find any enriched Biological Process Gene Ontology Terms. Among all the enriched terms, some of the most interesting ones because they are directly related to cancer were cell differentiation, positive regulation of cell proliferation, cell migration, cell division and cell death.

We used the pathview R package to visualize the results. In the maps, the colour scale represents the logFC of the differentially expressed genes, being green a negative logFC, or downregulation, and red a positive logFC, or upregulation. We started by plotting the KEGG cell cycle pathway (**appendix figure B.3**). In it we can see that mainly cyclins, and also the transcription factor E2F1,2 and 3 in the pathway show a significant increase in their expression. On one hand, cyclins are proteins involved in the regulation of the progression through the different phases of the cell cycle. On the other hand, E2F1, 2 and 3 are Transcription Factors whose DNA-binding site is in the promoter of a lot of genes involved in cell cycle regulation and DNA replication.

In the P53 signaling pathway (**appendix figure B.4**), not many of the nodes presented an important change of their expression. We see again how some cyclins and cyclin-dependent kinases (CDKs) show an increase of their expression. We also observe a significant increase in the expression of p14ARF, which acts as a tumor supressor gene by inhibiting MDM2, a known inhibitor of P53.

If we focus on the PI3K signaling pathway (**appendix figure B.5**), we see an underexpression of tyrosine kinase receptors (RTK) and growth factors. We also observe that there is an important upregulation of extracellular matrix related genes (ECM). We also see an increase in the expression of eIF4E, a protein involved in protein synthesis promotion that is part of the mTOR signaling pathway. It has been described that in cancer protein synthesis is usually increased (Vogt 2001).

Next, we will focus on the MAPK signaling pathway (**appendix figure B.6**). Again, we see an decrease in the expression of growth factors and RTKs. We also observe an increase in the transcription of TGF-β, a protein known to regulate cell proliferation and differentiation.

We can study the Breast Cancer KEGG network (Figure 4.4) to get an overview of the alteration state of the networks we have mentioned. In it we see how EGFR (Epidermal Growth Factor Receptor), which is a Tyrosine Kinase Receptor, shows a decrease in its expression. We can also see how Shc, a protein involved in the transduction of the signal originated by the binding of EGFR and its ligand, also seems to be unerexpressed. Another receptor that seems to show a downregulation of its expression is Frizzled, a receptor of the Wnt proteins. If we look at the genes showing an upregulation of its transcription, we can see E2F genes, which we mentioned previously, and Delta, the ligand of the receptor Notch.

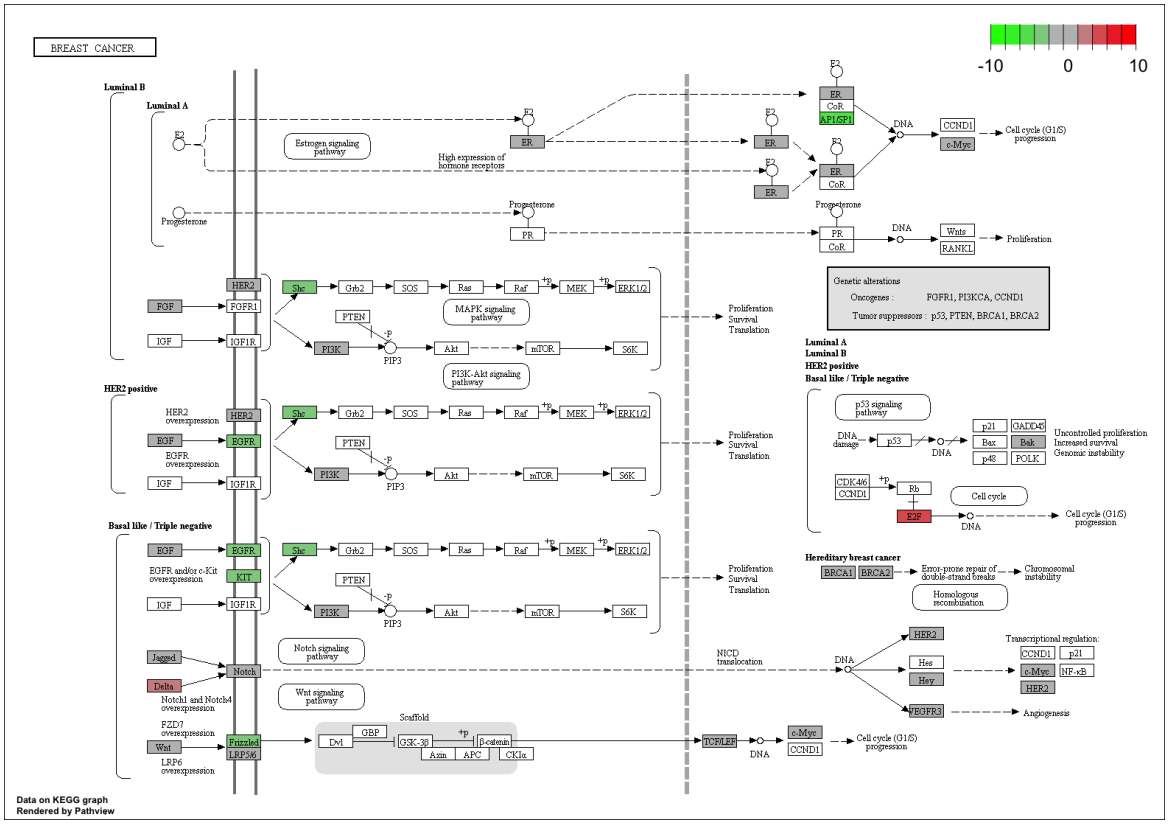


Figure 4.4: Breast Cancer KEGG Network overlapped with the results of the transcription data. Red nodes represent overexpressed genes and green nodes represent underexpressed genes. The intensity of the colour represents the logFC of the expression change.

In summary, the transcription data let us see how cell cycle might be promoted by the increaseof the transcription of several cyclins and CDKs. We have also observed how growth factors and RTKs seem to be underexpressed, which could be the result of a possible mechanism of downregulation if there was a big amount of incoming signal from the pathways downstream of these receptors. Another feature that we have observed is an increase in the expression of eIF4E, which is involved in the initiation of mRNA translation. We have also observed an alteration in Frizzled, a member of the Wnt signaling pathway. Hence, in the transcription data we have observed the biggest amount

of changes in the cell cycle pathway, where the alterations were observed in nodes with key roles.

4.2 Methylation

4.2.1 Exploratory Analysis

In an initial check of the data, we saw that there was a considerable amount of missing values in the data (about 18% of the values in the original data). Before going further in the exploration, we removed all rows containing missing values. We started by using the TCGAbiolinks package to create a boxplot containing the mean methylation values for each of the conditions (**Figure 4.5**). There, we could see that it appears that the mean methylation values for tumor samples were higher than the normal samples. Also, the methylation values seem to be more dispersed in the tumor samples than in the normal samples. These differences will be further assessed statistically in the following sections.

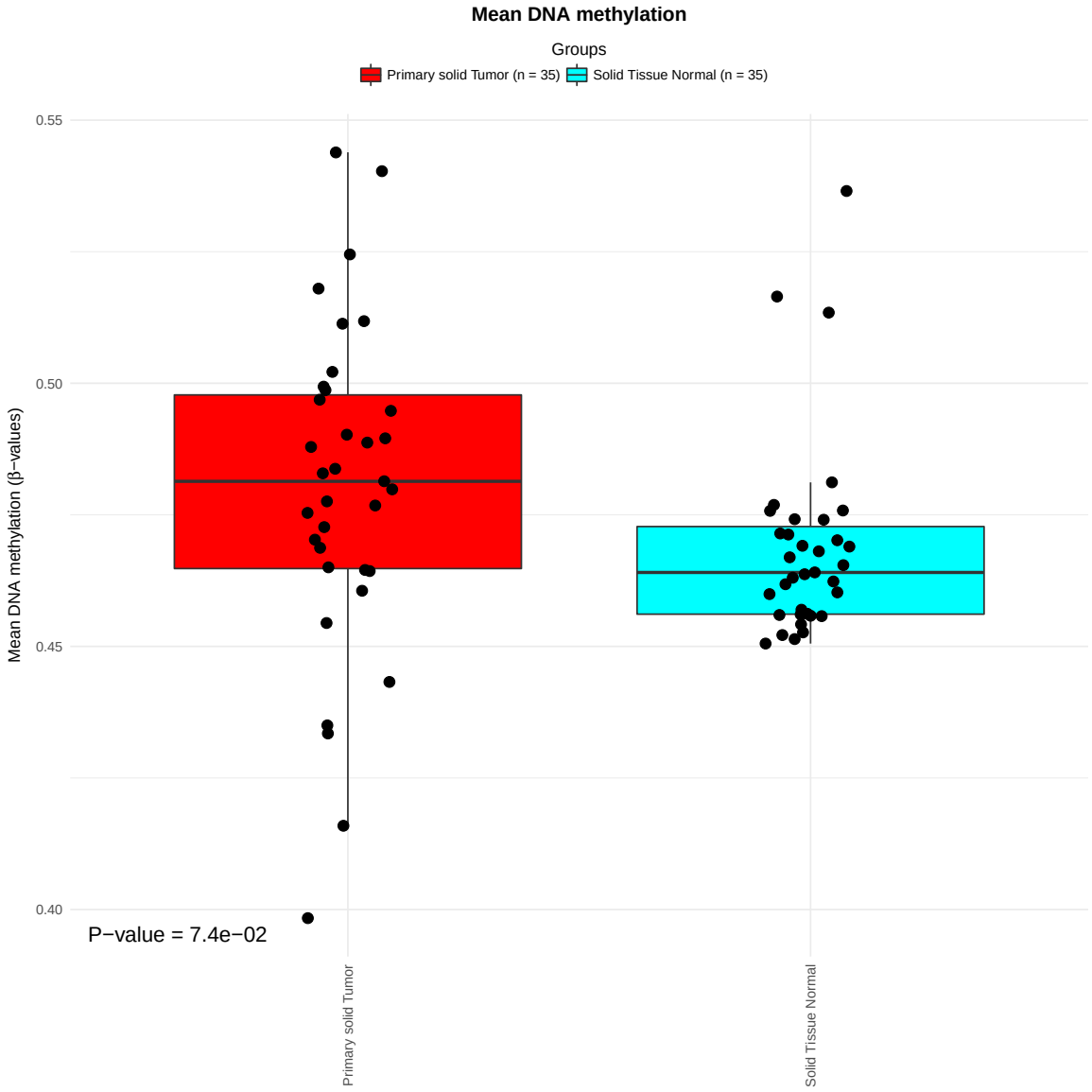


Figure 4.5: Mean methylation values for all probes in Tumor and Normal samples. The red boxplot represents tumor samples, while the blue one represents normal samples

Next, we wanted to see if there was any clustering when we applied a Principal Component Analysis on the Methylation data. The results of this PCA can be seen in **figure 4.6**. In it, we

can see two important patterns. First, we see that the PC1 separates normal and tumor samples. Second, we also see that the cluster containing the normal samples is much more compact than the one containing the tumor samples. The latter might be an indication of the tumor samples showing much more variability in the epigenetic landscape, while the normal samples could be much more stable epigenetically. The latter has been observed before in previous studies (Hansen et al. 2011).

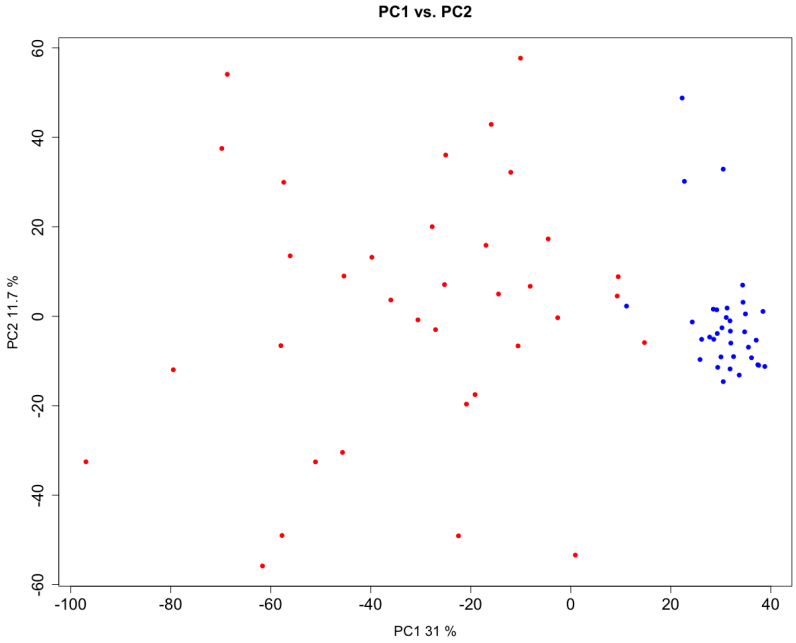


Figure 4.6: PC1 vs. PC2 of the Principal Component Analysis of the methylation data: red dots represent tumor samples, while blue dots represent normal samples.

Finally, to get an idea of the distribution of the β-values, we plotted the density of β-values for 10000 randomly picked probes. For reproducibility purposes, the set.seed(444) R command was used. The resulting plot can be found in **figure 4.7**.

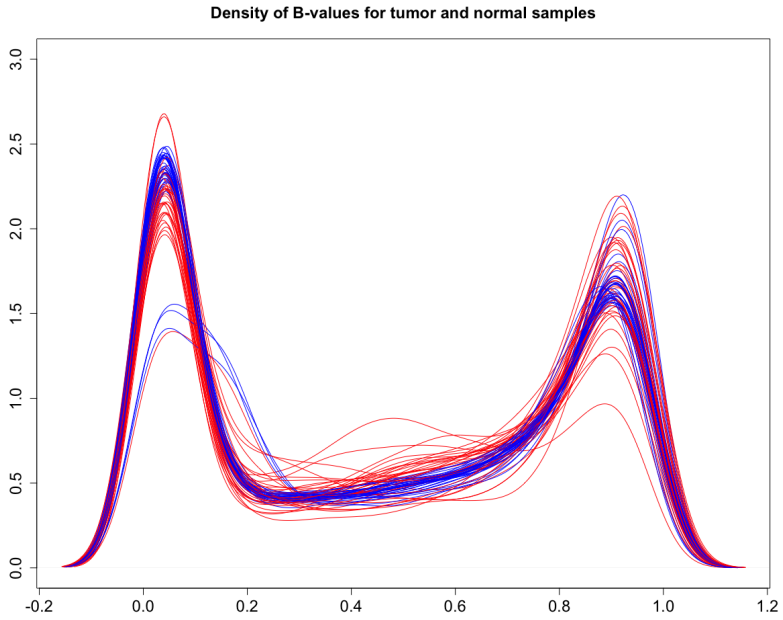


Figure 4.7: Density of β-values of 10000 random probes for each sample: red lines represent tumor sample, while blue lines represent normal samples.

In it, we can see that the highest density of values is gathered around 0 or 1, having an intermediate amount of probes showing values in between. If we look more closely, it seems that normal samples show a more similar distribution of the picked β-values than tumor samples, which seem to show a higher variability. It is important to note that this comments need to be tested statistically.

4.2.2 Differential Methylation Analysis

The analysis to assess the presence of differentially methylated probes across conditions was performed as described in the Material and Methods chapter. We found a total of 2114 differentially methylated probes. From all those probes, we selected the ones being the closest to the Transcription Start Site (TSS) of protein coding genes. In (Jiao, Widschwendter, and Teschendorff 2014), the authors observed that the probes mapping within 200 bp of the TSS were the one with a higher impact on the transcription of the associated genes. This left us with a total of 412 differentially methylated probes, where 381 of them were hypermethylated in tumor samples respective to normal, and 31 were hypomethylated.

Next, an enrichment analysis was performed to find enriched Gene Ontology terms in the list of genes showing a differential methylation in the sequences proximal to the TSS. Among all the enriched terms we found cell differentiation, cell fate specification, cell development, cell motility, localization of cell, regulation of response to nutrient levels, cell migration and negative regulation of cell differentiation. Once more, these terms are directly related to cancer.

Finally, we wanted to visualize the same KEGG pathways assessed previously to see if any of the genes in them where showing important changes in their methylation levels. The images for this can be seen in the **appendix figure B.7** (Cell Cycle Signaling Pathway), **appendix figure B.8** (P53 signaling pathway), **appendix figure B.9** (PI3K signaling pathway) and **appendix figure B.10** (MAPK signaling pathway).

In the cell cycle pathway (**appendix figure B.7**), only one node showed a significant change in its methylation value. In it, we can see that only MCM genes seemed to be hypermethylated. One of the functions of these genes is to regulate the initiation of genome replication. The same happened with the P53 signaling pathway, which did not show alterations in the nodes with key roles either. The network overlapped with the results can be found in (**appendix figure B.8**).

In the PI3K signaling pathway (**appendix figure B.9**) we can see that the promoters for some genes encoding for growth factors, as well as promoters for genes encoding for Extracellular Matrix proteins seem to be hypermethylated in the tumor samples. Also, TCL1, a protein involved cell proliferation promotion, seems to show a hypermethylation of its promoter.

Finally, we will focus on the MAPK signaling pathway (**appendix figure B.10**), where we can see that PKC seems to show a hypermethylation in its promoter. The function of this gene is to regulate cell proliferation and differentiation, among many others. We also see that MLK3, also known as MAP3K11, seems to show an hypermethylation of its promoter. This gene is also involved in cell proliferation regulation.

We will visualize the Breast Cancer KEGG network (**Figure 4.8**) overlapped with the results from the methylation data. The only genes for which we find changes in this network are FGF and IGF, both being growth factors.

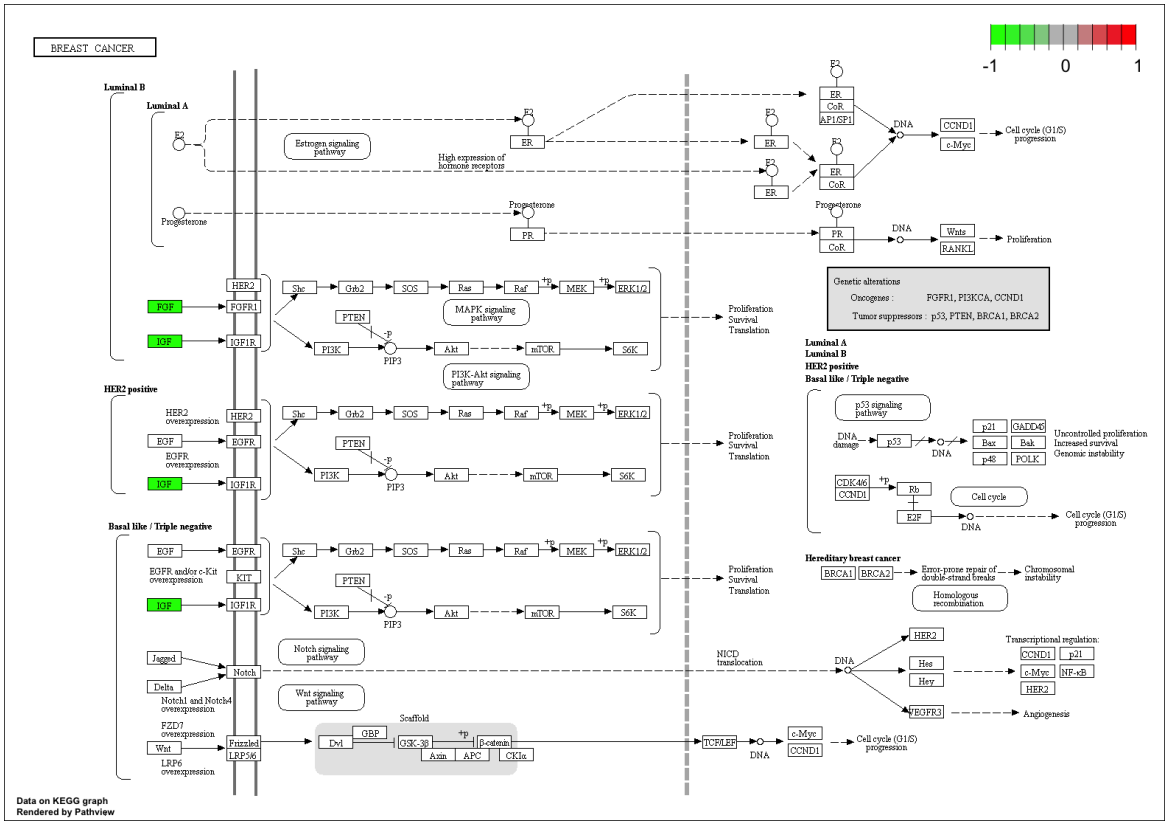


Figure 4.8: Breast Cancer network from KEGG overlapped with the results of the methylation data. Red nodes represent hypermethylated genes, while green nodes represent hypomethylated genes.

In summary, we have not observed a lot of changes in the methylation of the genes in the cell cycle or P53 signaling pathways. However, the PI3K and MAPK signaling pathways seemed to show some more affected nodes when we compared normal and tumor samples. It is known that an effect of hypermethylation in gene promoters is the repression of transcription of the affected genes. On one hand, in the transcription data we had seen that growth factors showed a decrease in their transcription, while in the methylation data we saw that some of those genes were hypermethylated. On the other hand, we had also observed a decrease in the transcription of RTKs, but we have not found a hypermethylation of its promoter. There could be several reasons for this result. First, the downregulation of RTK is not caused by a hypermethylation of its promoter. Second, we excluded the results of its probe when we applied the filtering of the results. Third, there were not probes designed to assess methylation events on that particular sequence.

4.3 CNVs

4.3.1 Exploratory Analysis

To explore the distribution of the CNV data we decided to check the density of the segment mean values for normal and tumor samples. Thus, in **figure 4.9** we can see the segment mean values for the tumor (red) and normal (blue) samples. If we look at it, we can see that the biggest density of values is around zero, which is what we would expect. However, if we look more closely, we will see that tumor samples seem to show a higher density than normal samples in values different from zero for the segment mean. As explained above, segment mean values higher than zero are interpreted as amplifications, and segment mean values lower than zero can be interpreted as deletions. Looking at this plot might be indicating that in the tumor samples we have higher copy number variations in tumor samples than in normal samples. This, however, would have to be tested statistically.

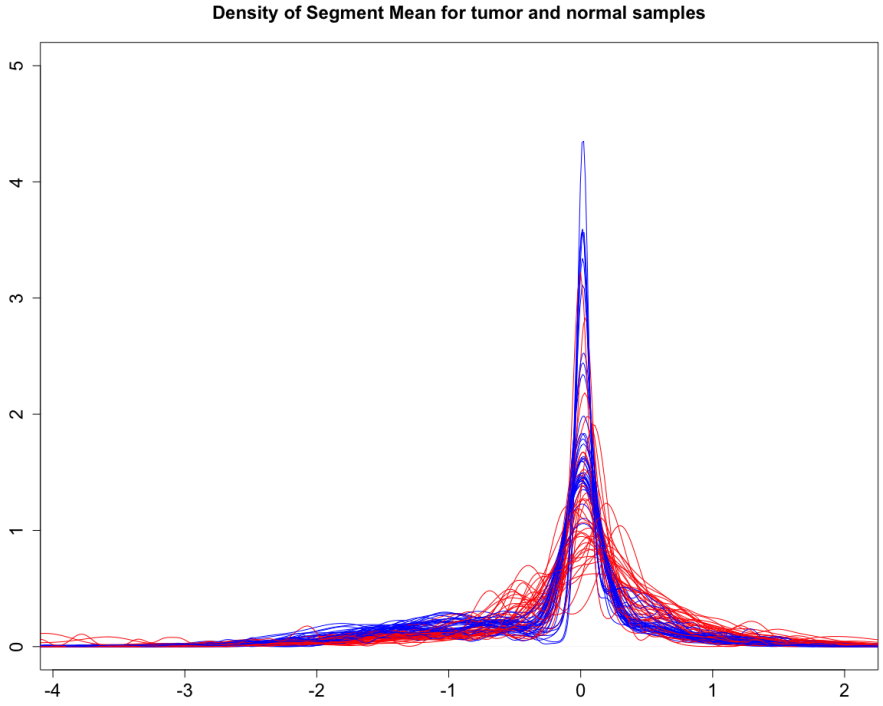


Figure 4.9: Segment Mean density for normal and tumor samples. Red and blue lines represent tumor and normal samples, respectively.

4.3.2 Recurrence analysis

Gaia was used to detect recurrently aberrant regions in our samples. The analysis resulted in 12611 different genes that seemed to be recurrently affected by some structural aberration.

Based on those, we realized a Gene Ontology Term (Biological Process) enrichment analysis with the GOstats package. Among the enriched Gene Ontology Terms, we highlight cell motility, as it can have an implication in migration of cancerous cells.

Finally, we also visualized the studied KEGG pathways to see the effect of potential CNVs affecting nodes in the Cell Cycle (**appendix figure B.11**), P53 (**appendix figure B.12**), PI3K (**appendix figure B.13**) and MAPK (**appendix figure B.14**) signaling pathways.

If we start by focusing our attention on the Cell Cycle signaling pathway (**appendix figure B.11**), we can see that a lot of cyclins and CDKs seem to show recurrent deletions. This is a bit against the information that we had seen previously, where it looked like we had higher levels of

transcription of these molecules. However, we also see that some of the cyclins and CDKs (cyclinD and CDK4 and 6) seem to show recurrent amplifications.

If we now look at the P53 signaling pathway (**appendix figure B.12**), we see that there does not seem to be a huge alteration in the network around P53, although some of the genes whose transcription is regulated by P53 and their downstream targets seem to be altered.

Let's now move to the PI3K signaling pathway (**appendix figure B.13**), where we can observe that a lot of the genes seem to be affected by one of the two kinds of recurrent aberrations. Among the recurrently deleted genes, we see GSK3 (involved in glycogen synthesis). We also find genes important in the pathway such as class IA PI3K, or FOXO. Some genes from the mTOR pathway, like eIF4B or S6 seem to also show recurrent deletions in our samples. These genes are known to be involved in the regulation of protein synthesis promotion. If we now focus on the recurrently amplified genes, we can see genes such as SOS (part of the MAPK signaling pathway) or the transcription factor CREB, among others.

Finally, we will focus on the MAPK signaling pathway (**appendix figure B.14**). Here, we can see that some of the recurrently deleted genes are RTKs or Ras, which are part of the classical MAPK pathway. These results would go against what we would expect, since these genes are known to be promoting cell cycle promotion in cancer. We observe amplifications in genes such as SOS or RAFB, which are an important part of the transduction of signaling downstream of the classical MAPK signaling pathway.

We will visualize the Breast Cancer KEGG network (**Figure 4.10**) to get an overview of the results. Some of the genes that seemed to find recurrent deletions were FGFR, FGF or EGF, which are RTKs and their ligands. We can also see taht Wnt and a lot of components of its pathway seemed to show deletions. We observe deletions in PI3K, and also in the Estrogen Receptor (ER). If we look at the amplification side, we can see CDKs and SOS, as we mentioned previouslu. We can also see Frizzled, the receptor of Wnt proteins.

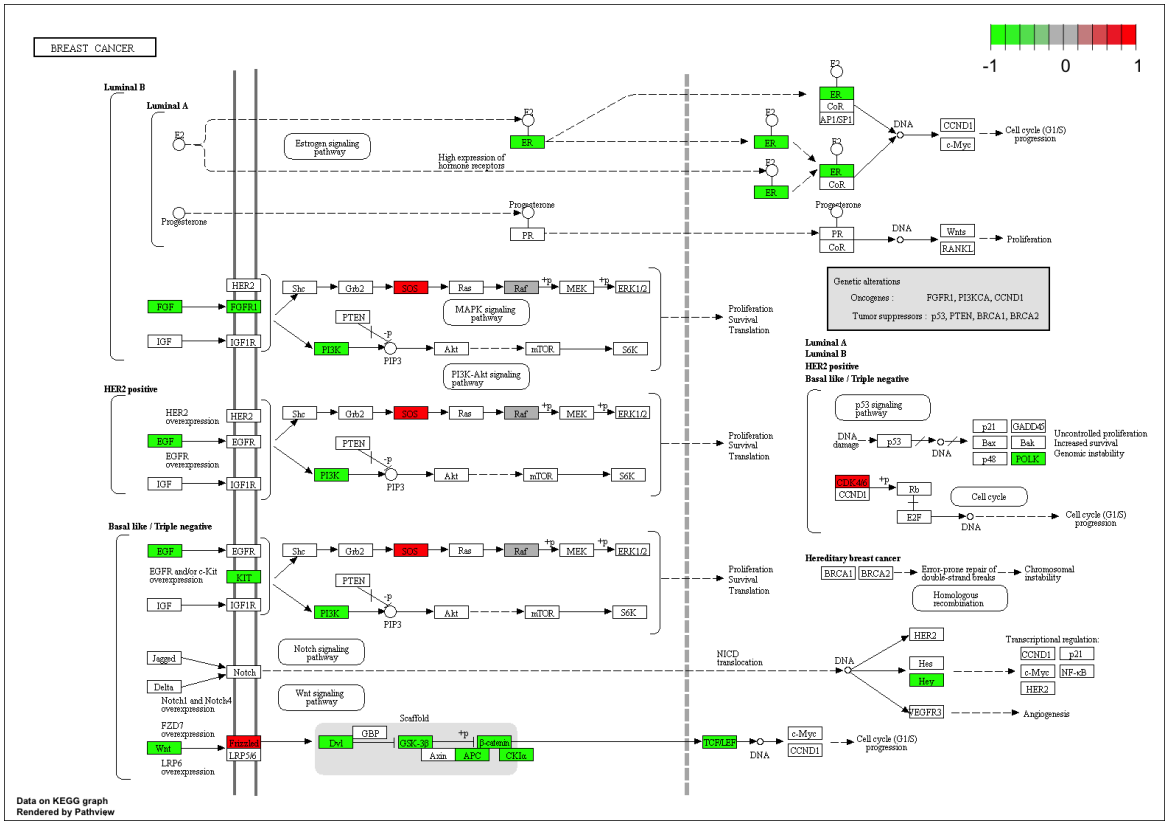


Figure 4.10: Breast Cancer network from KEGG overlapped with the results from the recurrency analysis. Red nodes represent genes affected by recurrent amplifications, while green nodes represent genes affected by recurrent deletions.

To summarize, we have observed that there seem to be a lot of genes recurrently altered from a Copy Number Variation point of view. However, we obtained a total of 12611 genes, so it was to expect that we would find such an amount of affected genes in the networks. We observed many more recurrent deletions than amplifications in the networks. We have observed how key nodes of the PI3K and the MAPK signaling pathways seemed to show recurrent alterations. For instance, we have observed deletions affecting PI3K, FOXO, RAS or RAF among others. We have also observed alterations in a big proportion of the components of the Wnt pathway, where all but one of the affected genes showed deletions.

If we focus again on the growth factors and RTKs, we can see that they also seem to be affected by recurrent deletions. We had seen how there was a decrease in the expression of both GFs and RTKs. We had also observed a hypermethylation of GFs, while we had not found a change in methylation of RTKs. However, we have observed a recurrent deletion that affects RTK genes, which could be an explanation of the decrease in its expression.

It is important to note, however, that we can't imply any causation from the results we have gotten. To test these hypotheses we would need an approach suited to this kind of questions.

4.4 DNA-seq

4.4.1 Exploratory Analysis

Overall, the samples showed a total of 1827 different mutations, where 191 were frame shift or in frame insertions/deletions and 1636 were point mutations. On one hand, the most frequent type of insertions/deletions were frame shift deletions, with a total of 122 mutations in our samples. On the other hand, the most common point mutations were missense mutations, with a total of 1460 (about 80% of the total), followed by nonsense mutations, with a total of 113 (about 6% of the total). A summary of this information can be found in **table 4.1**.

Mutation Type	Number	Mean	Median
Frame shift deletion	122	3.5	1
Frame shift insertion	47	1.4	1
In frame deletion	19	0.5	0
In frame insertion	3	0.08	0
Missense mutation	1460	42.941	30
Nonsense mutation	113	3.3	2
Nonstop mutation	3	0.08	0
Splice site mutation	58	1.7	1
Translation Start Site	2	0.05	0
Total	1827	53.7	34.5

Table 4.1: Summary of number, mean and median of mutation types across samples.

We also checked for the annotation of the impact of the different observed mutations in our samples (**Table 4.2**). We could see that 1481 of the mutations were annotated to have a moderate impact, while 345 were annotated to have a high impact.

Mutation Impact	Number
HIGH	345
MODERATE	1481
MODIFIER	1

Table 4.2: Summary of mutation impacts in our samples.

In **figure 4.11** we can get a more visual summary of the mutation state of our samples. In it, we can see the mutation types and the relevance of each type (top part and bottom middle panels). We see that missense mutations are by far the most common type of mutation (top left and bottom middle panels), being $C > T$ transitions the most common ones (top right panel). In the top middle panel we can also see that Single Point Mutations are the most common type of mutation. In the bottom left panel we can see the amount of mutations per sample. Finally in the bottom right panel we see the most altered genes, being TP53 and PIK3CA the first two most altered ones. On one hand, TP53 has been widely studied as one of the most frequently altered genes in breast cancer (Duffy, Synnott, and Crown 2018). Its role is to block cell cycle progression if it detects that there are signs of DNA damage, allowing the cell to repair these damages if possible or sending the cell to apoptosis if reparation is not possible. Loss of function mutations in this gene are known to be related to cancer progression. On the other hand, PIK3CA is one of the most important genes in the PI3K signaling pathway, which has been described to be involved in cancer progression and can be targeted in cancer therapy (LoRusso 2016).

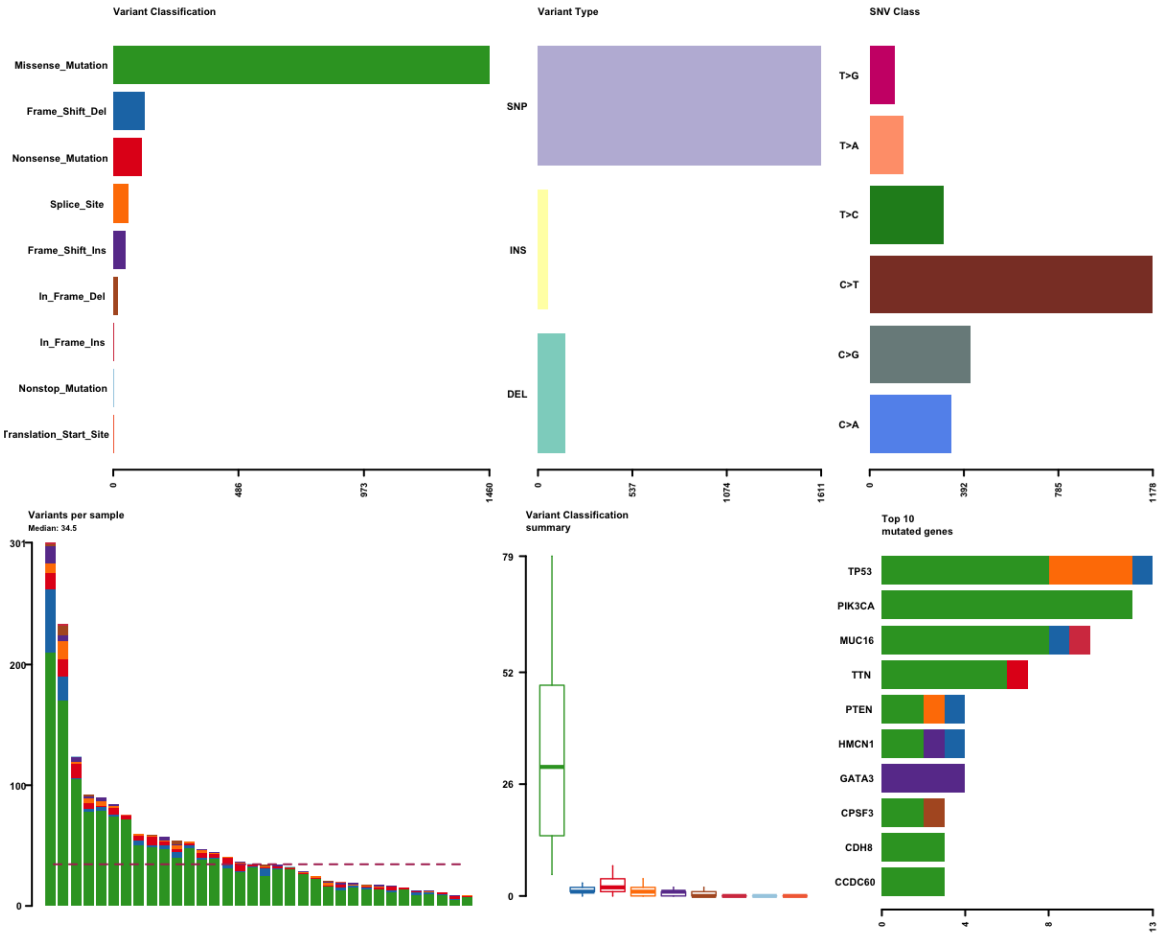


Figure 4.11: Mutation summary over the tumor samples compared to the normal samples. Top left: types of mutation detected and their abundance. Top middle: abundance of point mutations, insertions and deletions. Top right: types of single point mutations and their abundance. Bottom left: number of mutations and their type per sample. Colour code represent mutation types in the same colour code than in the top left panel. Bottom middle: boxplots with the distribution of the different mutation types, where colours are in the same colour code as in the top left panel). Bottom right: top 10 most mutated genes across samples. Colour codes represent the type of mutation observed.

To get an idea of the domains where the mutations were being found in the genes, we created a lollipop plot for TP53 (**Figure 4.12**) and PIK3CA (**Figure 4.13**). In these plots, the x axis contains the amino acid sequence, where the different coloured boxes represent known functional domains of the protein. The y axis represents the amount of detected mutations in our samples in a specific amino acid position.

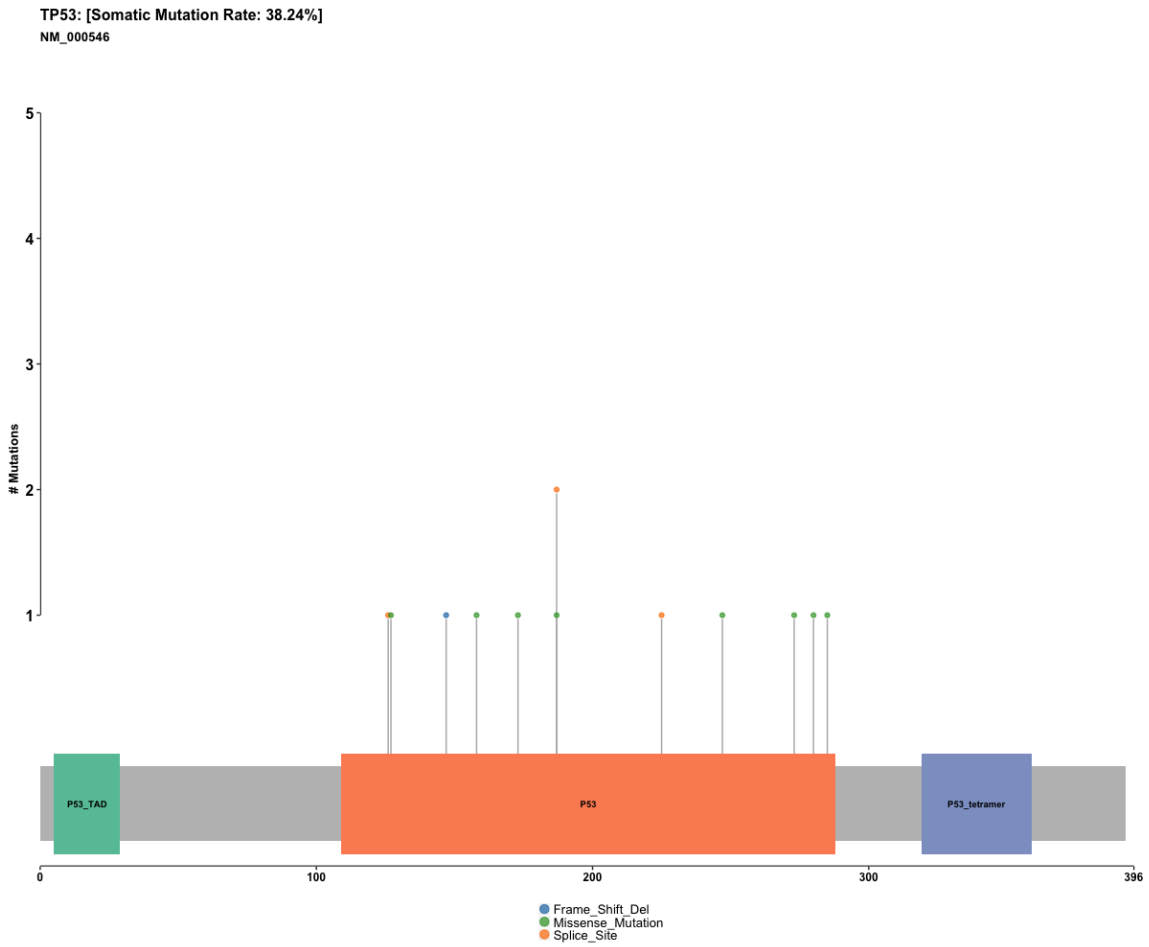


Figure 4.12: Lollipop plot of P53. The different domains are represented as coloured boxes, while the number and type of mutations are represented in the y axis.

If we look at the lollipop plot for TP53 (**Figure 4.12**), we see that the mutations are all gathered in the P53 domain. The mutation types observed in this domain are Frame shift deletions, Missense mutations and Splice Site mutations. The P53 domain of this protein has a DNA-binding function, allowing P53 to bind to the DNA and develop its function as a Transcription Factor. Hence, mutations in it can potentially have a big effect on the function of this protein.

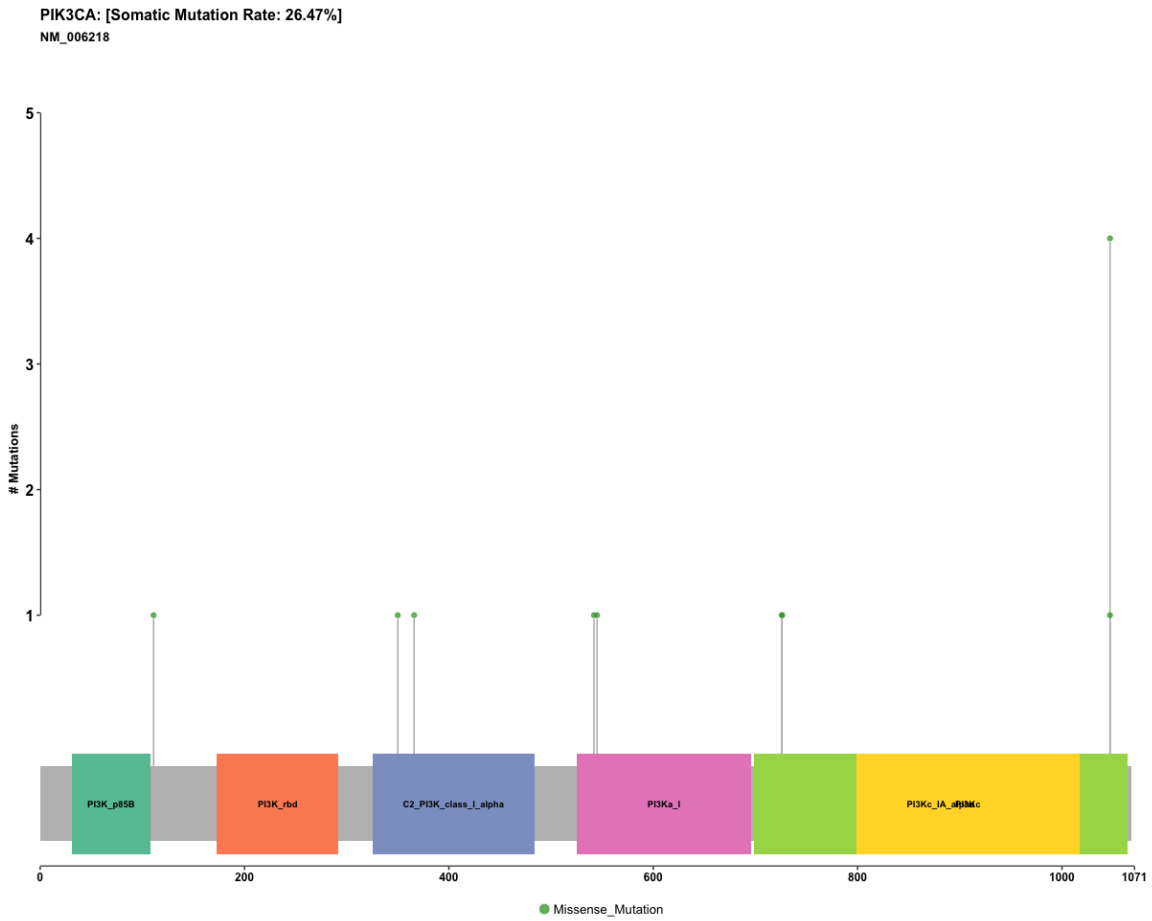


Figure 4.13: Lolliplot plot of PI3KCA. The different domains are represented as coloured boxes, while the number and type of mutations are represented in the y axis.

On the other hand, in the lollipop plot for PIK3CA (**Figure 4.13**) we can see that mutations are spread around the C2 domain (involved in the interaction of PI3KCA with the membrane, where it will develop its function), PI3Ka_I (whose role is unclear acoording to the information stored in PFAM (Finn et al. 2016)) and PI3_PI4_kinase domains (involved in the catalytic activity of the protein). We only observe missense mutations in this case, and the highest amount of mutations is observed in the catalytic domain. Gain-of-function mutations in this domain could increase the activity of this enzyme, which has been seen to be related to cancer progression (Kandula et al. 2013).

4.4.2 Analysis

As mentioned in the Material and Methods section, the analysis of the DNA-seq data is much more descriptive than the rest of data types. Based on the computed score for each gene, we observed the maps for the selected KEGG pathways using the pathview package. In this map, values can range from 0 to infinity, and more reddish nodes represent genes that are more frequently altered in our samples. If we start by the Cell Cycle signaling pathway (**appendix figure B.15**), we see it seems that an important amount of nodes are showing mutations in our samples. Among those, we could highlight some cyclins and CDKs, which have a main role in regulating the cell cycle progression, Rb (who is known to interact with E2F1 and promote cell cycle arrest), APC/C (involved in the transition from anaphase to metaphase in the mytothic cell cycle) and P53, which we will comment in the next figure.

If we now observe the P53 signaling pathway (**appendix figure B.16**) we see that the most altered gene is P53 itself. We had seen that those mutations were mainly in the DNA-binding domain, which could potentially importantly alter its activity. We also observe that PTEN, a protein known to be doing the inverse process of PI3K, is also showing to have some mutations in our data.

Next, we will focus our attention on the PI3K signaling pathway (**appendix figure B.17**). We see that this pathway also has a considerable amount of genes that show mutations in some of the samples. Of special interest are the tyrosine kinase growth factor receptors (who initiate signal transduction after binding to some growth factors), PI3K itself, which as mentioned previously has been observed to be involved in cancer progression, PTEN (which has a function in the opposite direction of PIK3CA) and AKT (one of the most important proteins in the signal transduction of the PI3K signaling pathway). Finally we also see that mTOR (which has been targeted in cancer therapy by the use of rapamycin) is seeming to be frequently mutated in our samples.

The last pathway we are going to focus our attention on is the MAPK signaling pathway (**appendix figure B.18**). Apart from the previously mentioned genes, we can see that MEKK1 and MKK4, also known as MAP3K1 and MAP2K4, seem to be frequently mutated in our samples. These genes are known to be important in signal transduction in the MAPK signaling pathway, which is targeted in therapy and has been studied in cancer research.

We can observe the Breast Cancer network from KEGG overlapped with the results of the mutation in (**Figure 4.14**). Apart from the ones we mentioned already, we observe mutations in some of the components of the Wnt Pathway such as Wnt, Frizzled, APC or TCF/LEC.

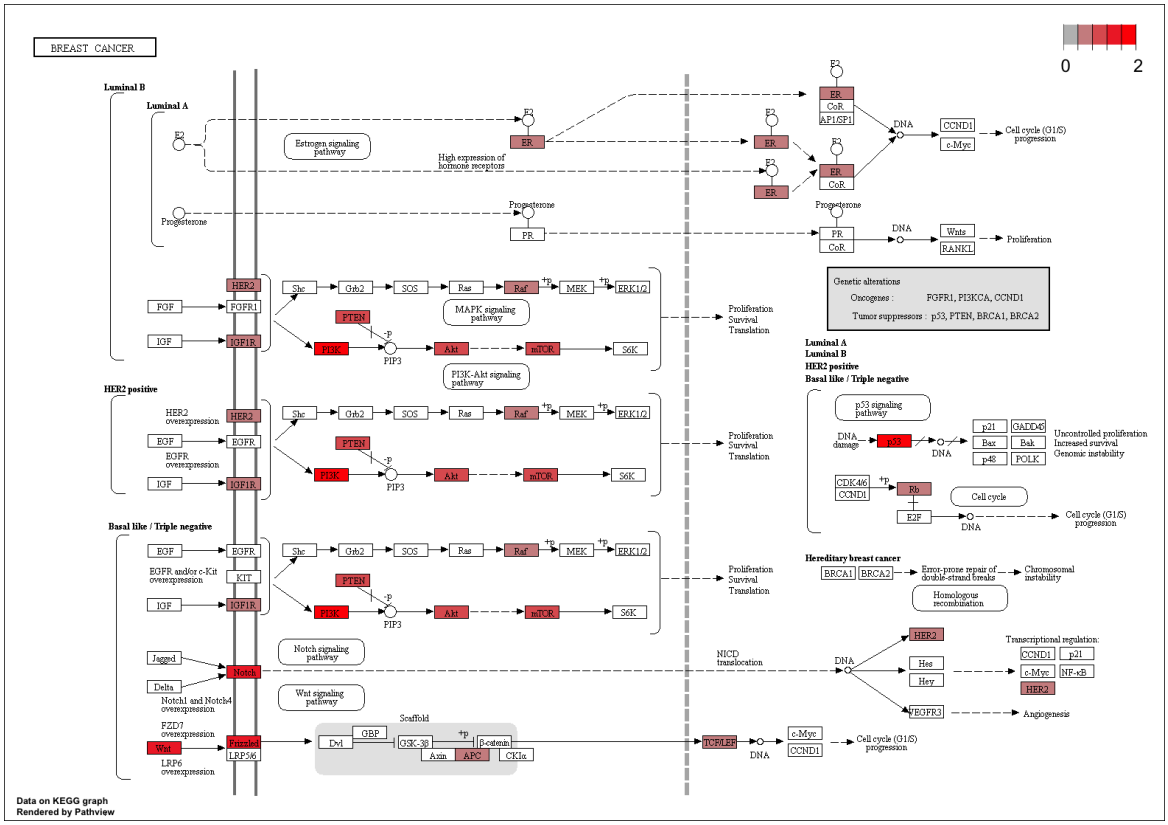


Figure 4.14: Breast Cancer KEGG network overlapped with the results from the mutation. Here, red nodes represent genes that show mutations in our samples.

To summarize, we have observed that a considerable number of the genes in the studied networks showed mutations in our samples. Among those, we could highlight P53, which had not been observed to be altered until now. This could be an indication that, in our samples, mutations are the most common mechanism that affect the function of P53 and block it from stopping the cell cycle. We

have also observed mutations in PI3K, PTEN, AKT, RTKs or RAF, which are key genes in their respective signaling pathways. We have also seen genes from the Wnt Pathway showing a high score. Some of those are Wnt, Frizzled, APC and TCF/LEF.

4.5 Integration

Once we obtained a score for the different genes, we visualized the results with the pathview package. As mentioned in the Material and Methods chapter, lower scores indicate genes that show alterations in a lower number of omics data types, while higher scores indicate genes that show alterations in a higher number of omics data type.

We will start by observing the Cell Cycle signaling pathway (**Figure 4.15**). There, we can see that some of the genes that seem to show a higher alteration are APC/C, which as we said is involved in regulating the transition to anaphase, and several cyclins and CDKs. After looking at the different individual results (appendix figures B.3, B.7, B.11 and B.15) we can see that the data types that contributed the most to the integrated scores are transcriptomics, copy number variations and mutations. We saw how in the methylation only one node presented an altered type. This might be an indication that methylation is not a common mechanism involved in alterations of these genes.

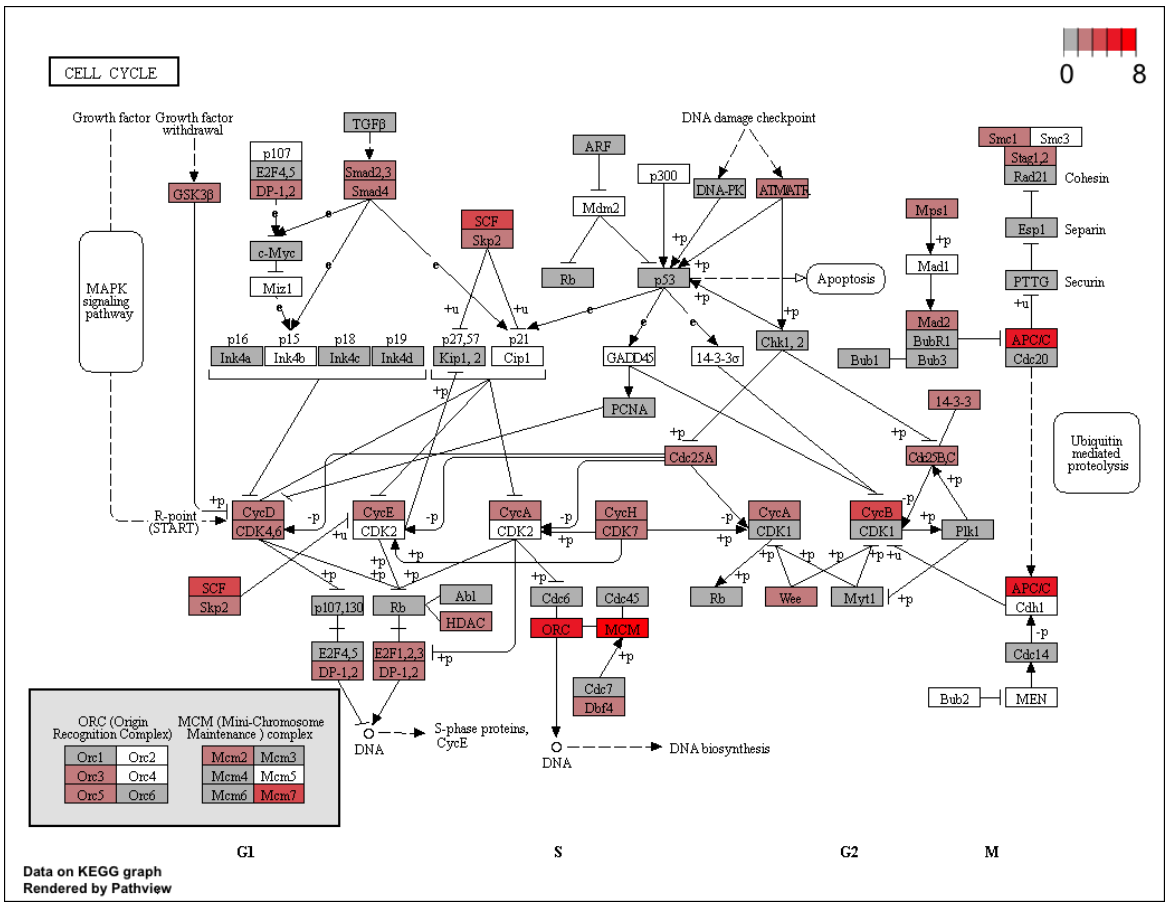


Figure 4.15: Cell Cycle pathway from KEGG overlapped with the final score computed for each gene. Here, gray nodes represent genes that show alterations in a lower amount of omics data type, while red nodes represent genes that are altered in different types of omics.

In the P53 signaling pathway (**Figure 4.16**) we can see that the more alterations were observed downstream of P53 than in P53 itself and the proteins around it. If we check the individual results (appendix figures B.4, B.8, B.12 and B.16), we can see that there was a consensus on some nodes. We saw how each different data type detected alterations in different genes in the network, but genes like P53, which is known to be highly involved in cancer were not highlighted by the integration approach. As we mentioned, the only data type that detected alterations in P53 was DNA-seq. This might be indicating that, in cancer, the most common mechanism of alterations in P53 is mutation of important domains such as its DNA-binding site.

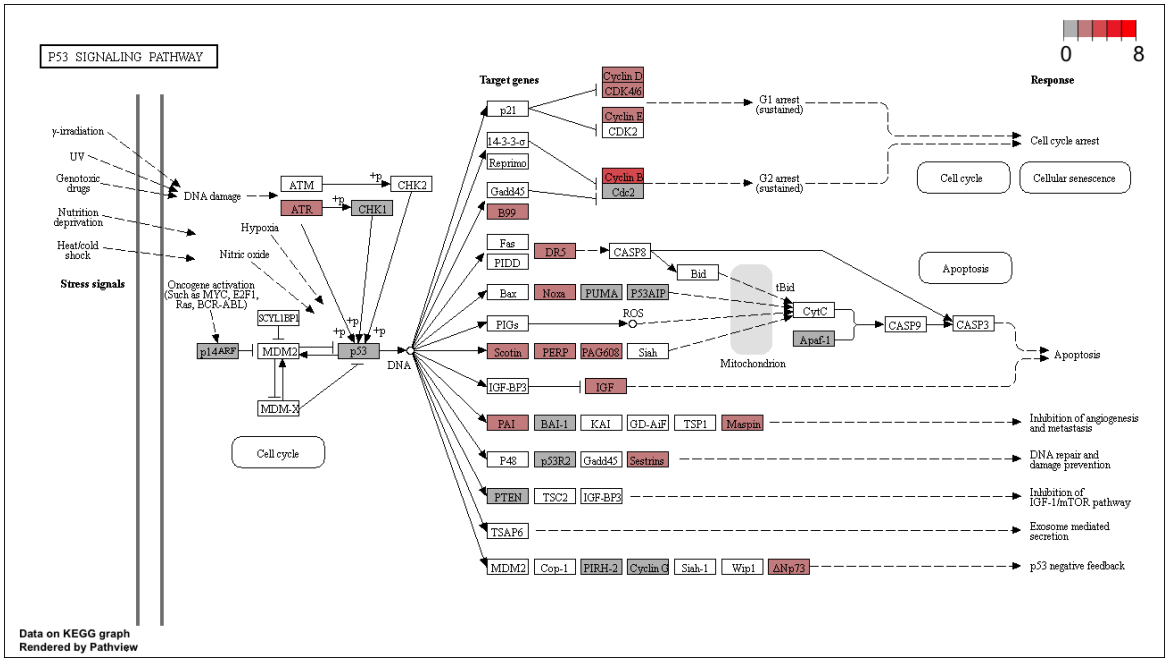


Figure 4.16: P53 pathway from KEGG overlapped with the final score computed for each gene. Here, gray nodes represent genes that show alterations in a lower amount of omics data type, while red nodes represent genes that are altered in different types of omics.

If we observe the PI3K signaling pathway (**Figure 4.17**), we will see that the highest amount of alterations was observed at the top part of the network. This could be suggesting that the upper elements of this signaling cascade are susceptible to more types of aberrations, while the components that are further down in the signaling cascade are more prone to suffer very specific types of alterations. An example of the first would be class IA PI3K, which seems to be highly altered in our samples, while an example of alterations in components that are further down in the signaling would be AKT, which only showed mutations in our samples.

We can also see that the mTOR signaling pathway, which is closely connected to the PI3K signaling pathway, seems to show some degree of alteration.

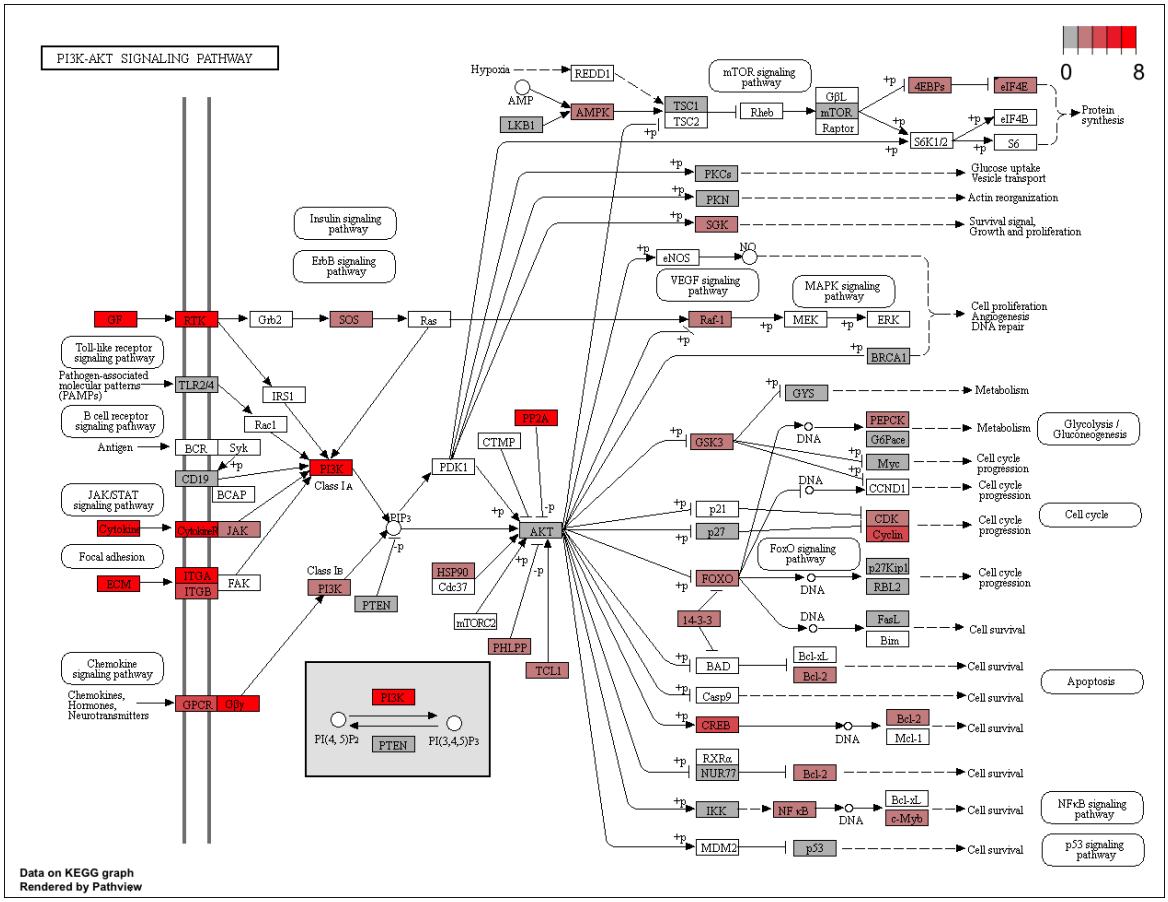


Figure 4.17: PI3K pathway from KEGG overlapped with the final score computed for each gene. Here, gray nodes represent genes that show alterations in a lower amount of omics data type, while red nodes represent genes that are altered in different types of omics.

When we look at the MAPK signaling pathway (**Figure 4.18**), we can see that a lot of nodes show at least some degree of alteration, where the ones showing the highest values are growth factors and RTKs, who are at the very beginning of the signal transduction of this pathway. We also see that some of the downstream nodes such as SOS, Ras, Raf and several MAPK show some degree of alteration. In this case, many of the most relevant changes were detected by mutation and copy number variation data types, which might be suggesting that these two mechanisms of alteration are the most relevant ones in the MAPK signaling pathway.

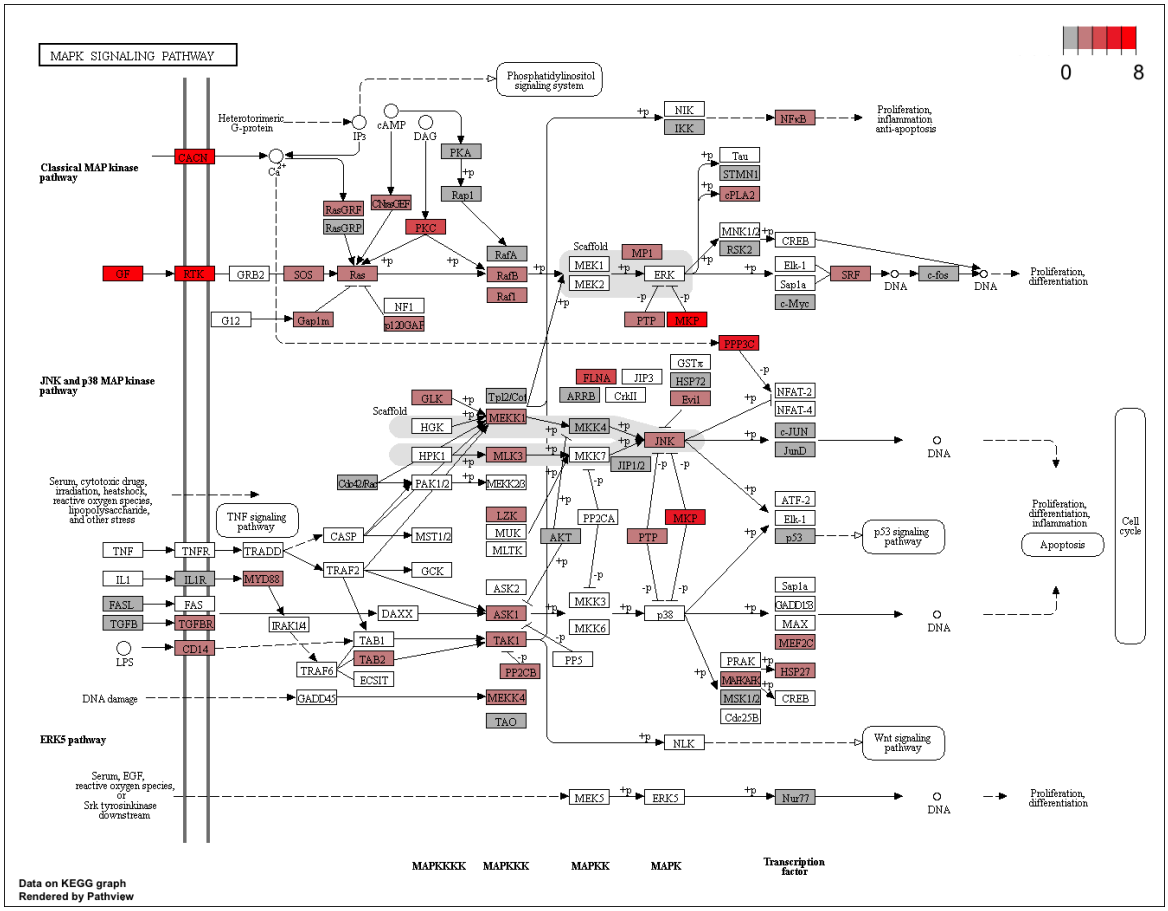


Figure 4.18: MAPK pathway from KEGG overlapped with the final score computed for each gene. Here, gray nodes represent genes that show alterations in a lower amount of omics data type, while red nodes represent genes that are altered in different types of omics.

Finally, we will look at the Breast Cancer network from TCGA (**Figure 4.19**), where apart from the previously mentioned observations we can also see that the almost all the components from the Wnt pathway seem to be highly altered. Also, this network contains different modules for the different defined subtypes of breast cancer (Luminal A, Luminal B, HER2 positive and Triple Negative). We see how nodes in all of these modules seem to show some degree of alteration, suggesting that the integrative approach is able to capture alterations in relevant mechanisms of Breast Cancer progression.

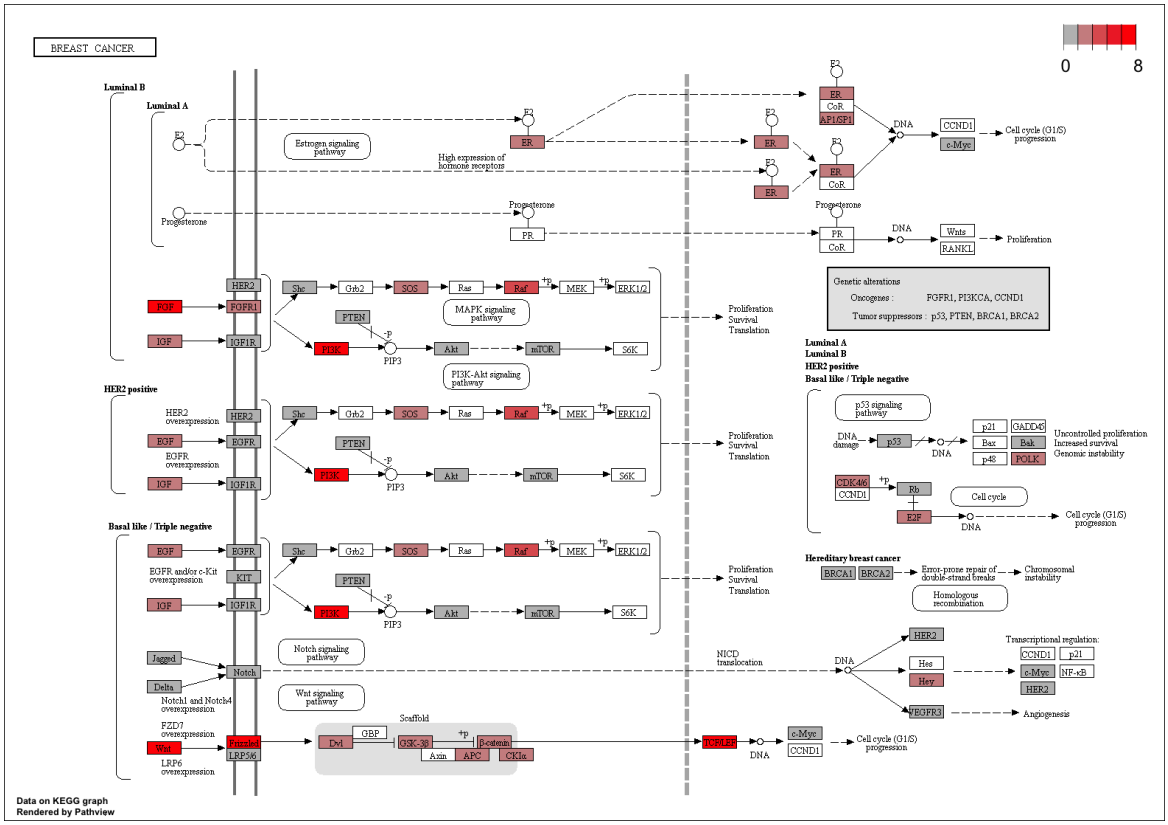


Figure 4.19: Breast Cancer pathway from KEGG overlapped with the final score computed for each gene. Here, gray nodes represent genes that show alterations in a lower amount of omics data type, while red nodes represent genes that are altered in different types of omics.

In summary, we have observed how some of the genes have stood out after integrating the several results, while some have become less important. Among the most altered genes in the integration approach, we observe APC/C, cyclins, CDKs, Growth Factors, RTKs, PI3K, RAF or PKC. After visualizing the integrated results with a Breast Cancer network from KEGG we have seen that some nodes in all of the three subtypes were showing alterations, where the most altered ones seemed to be PI3K, FGF, RAF, WNT or TCF/LEF.

Chapter 5

Discussion

Omics data integration can allow us to get deeper insights into the cause of alterations in genes of interest. Some studies have looked for biomarkers based on one type of omic data, transcription being the most widely used for this purpose. Although we might find that a gene is consistently underexpressed in a given condition, we would not be able to tell the cause of this change. The integration of several data types could help to discern genes that can show alterations through a lot of mechanisms from genes that are very frequently altered by only one of those mechanisms. This could help in understanding disease mechanisms, but might also be useful in the diagnosis or treatment selection processes. For instance, it would be interesting to apply the pipeline developed in this thesis to a data set that contained enough replicates of tumoral and normal tissues from a same patient. Since the data would come from a single patient, we might expect to find less sparsity in the alterations, which might allow us to highlight very specific cellular pathways. These pathways could then be considered in a targeted treatment.

In this work we have integrated several data types in order to see if we could get more insights into some networks that are known to be involved in cancer. We have used Breast Cancer samples from patients from which we had transcription, methylation, copy number variation and mutation data for both normal and tumoral tissues. We have used a scoring system that would allow us to identify highly altered genes, while it would not highlight those genes who only showed alterations in a lower amount of data types. We have also performed individual analyses to see how the landscape would change from the individual point of view to the integrated point of view. This has allowed us to see that some genes like P53, whose alteration is known to be very involved in cancer, only showed alterations in the mutations data, while it did not show alterations in any of the other types of data. We have also seen how some genes like PI3K seemed to be altered in more than one data type, making it one of the genes that stood out the most in the integrated maps. In the case of PI3K we have seen that the data types that contributed the most to the final integrated score were transcription, copy number variation and mutation. When looking at other genes we have seen how different types of omics data contributed more than others in the final score. These might be pointers indicating that some genes are prone to suffer a very specific type of alteration while others might be susceptible to a wider range of alterations.

The explanation of potential disease causing events from an omics point of view is a major challenge. From a transcriptomics point of view, an increase in the transcription might lead to higher concentrations of a protein with a function related to cancer progression. The opposite is also possible, where we would find that a decrease in the expression of a tumor suppressor gene might lead to lower concentrations of this gene, which may result in a lower tumor suppressing activity. In cancer samples, some genes might be more susceptible to show bigger changes in their expression in cancer because they are targeted by transcription factors that are part of cancer related pathways. Furthermore, they could also be more susceptible to show changes because they are in a part of the genome that is recurrently amplified or deleted in cancer, because they tend to accumulate mutations in their promoter or also because they tend to be hyper/hypomethylated in cancer.

If we look at the problem from a methylation point of view, we see on one hand that a hypermethylation of the promoter of a tumor suppressor gene might lead to the repression of its transcrip-

tion, which again would lead to a lower amount of tumor suppressing activity. On the other hand, a hypomethylation of the promoter of an oncogene might lead to an increase of its transcription, facilitating cancer progression.

If we considered the point of view of copy number variations, genes within a region that is recurrently amplified or deleted might be showing an increase or decrease in their expression, respectively. Another possibility would be that a copy number alteration resulted in the truncation of a gene that was in the boundary of the alteration. This truncation would lead to the production of a non functional protein.

The study of mutations in specific genes might allow us to find nucleotidic positions that are prone to cause gain-of-function or loss-of-function alterations. For instance, gain-of-function mutations in a gene involved in cell cycle promotion could have a key impact in cancer progression, while loss-of-function mutations in tumor suppressor genes might block these genes from stopping cell cycle promotion.

There are even further data types that could be considered, such as phosphoproteomics, or the study of micro-RNAs (miRNAs). The first would allow us to see if a protein is showing higher or lower levels of its phosphorylated form in a condition compared to another. This is important because the activity of an important number of the proteins in the signaling cascades we have studied is regulated by phosphorylation. The second type of molecule (miRNA) is a type of non coding RNA that is known to bind to specific mRNAs and promote their degradation. Thus, an increase of a miRNA who targeted transcripts of a tumor suppressor gene might lead to a decreased production of the protein encoded by that gene.

In the interpretation of both the individual and integrated maps, it is extremely important to take into account what was the meaning of the test that we applied. For instance, we saw that we obtained a total of 12611 genes that seemed to be recurrently affected after applying Gaia even after applying quite astringent parameters. Although it is known that cancer cells are subject to a high genomic instability, our results seem to show an extreme number of recurrently amplified/deleted genes.

We have observed that a considerable amount of genes in the studied pathways seem to show alterations from more than one data type perspective. Although this can be seen as a redundancy in data, it also allows us to highlight genes that are highly altered in our samples. By considering several points of view, we might also be able to consider more interpretations of the results of one of the data types. For instance, we might detect that a gene is showing a lower transcription in the tumor samples. If we saw that there had been a deletion of the region containing that gene, we might speculate that a cause for the decrease in that gene's transcription is that it got deleted from the genome. Other possibilities could be a hypermethylation of the gene's promoter, which has been identified as a potential gene silencing mechanism. Even more complicated hypotheses would be that an important gene showed a gain-of-function mutation which resulted in an increased activity of the pathway it belonged to. It is known that some mechanisms to regulate the activity of a pathway is the creation of feedback loops, where the activity of the pathway regulates the transcription of the genes involved in that same pathway. These type of hypotheses, although really complicated to test, would probably not be possible to investigate without considering different types of data.

Interestingly, we observed that genes being part of upper parts of a signaling cascade seemed to be highly altered, while genes that were more downstream in the cascade seemed to show more specific types of mutation. Although this is just an observation, it would be interesting to further study if there is any reason behind this observation or if it was just a coincidence.

When comparing results from individual analyses we saw that sometimes they were seeming to be coherent while sometimes they seemed to be contradictory. An example of this is that we found that some genes encoding for growth factors were showing an underexpression and a hypermethylation. These two results are coherent in the way that hypermethylation is a known mechanism that represses gene expression. However, when we studied the Copy Number Variation data we saw that those genes seemed to show recurrent deletions. If we had only considered the transcription and copy number variation data, we could hypothesize that the decrease in the transcription of those genes might be caused by the observed recurrent deletion. However, the interpretation of a gene showing a hypermethylation in its promoter when a deletion has been detected is more difficult.

It is important to point that the definition of the weight of each data type on the final score is

arbitrary, which means that different weights will lead to different results. In this work, we decided to assign an equivalent weight to transcription, methylation and copy number variation data because we had the data we required to perform a statistical analysis. However, to study the significance of mutations in our samples we needed access to the BAM files, which is restricted in TCGA. Thus, since we could not test the significance of mutations in genes we decided to lower the weight that mutation data would have in the final score.

Chapter 6

Conclusions

6.1 Learning during the development of the thesis

The development of this thesis has allowed me to get a better understanding of the analysis of different data types, as well as making me think about ways of integrating different results. I also had to put special efforts in the analysis of some of the data types, like Copy Number Variation data, with which I was not so familiar.

6.2 Changes in the original plan

Initially, miRNA data was also intended to be used. However, miRNAs are not included in the networks we have studied in KEGG. It is due to this fact that we decided not to include miRNAs in the final analysis.

6.3 Beyond this work

The purpose of this work was to create a pipeline that would allow us to see highly altered genes in known cancer related pathways. This has been accomplished by following the objectives described in Chapter 2 of the thesis. Although we have found that relevant genes in the studied pathways show alterations in the different data types, it would be interesting to test how this pipeline would perform if we had data from a single patient with a number of replicates big enough to ensure a minimal statistical power.

It would also be interesting to further study if some genes are targeted by very specific types of alterations, while others are targeted by more than one type. If this was the case, it would be interesting to study if a specific type of alteration has a significant effect in the response to a treatment. It might also be of interest to study if indeed genes that are in upper parts of a signaling cascade tend to be susceptible to more types of alterations than genes that are further downstream in the signaling pathway.

In this work we have used the probe closest to the Transcription Start Site of protein coding genes. However, there are approaches that use the average of a subset of probes annotated to a gene. Other options would be to consider a profile of probes to assess a methylation value to a gene. There is not a consensus on how to proceed in the assignment of a methylation value for a gene. It might be relevant to consider how robust are the results when we use different strategies.

We saw in the results from the recurrent Copy Number Variations analysis that we ended up with a total of 12611 affected genes. It might be important to test for recurrent aberrations with other tools to check the consistency of these results.

We wanted to also perform a statistical test to consider the signification of mutation events in our data. As explained in the Material and Methods chapter, we wanted to use the tool MuSiC to assess this. However, in order to use this tool we needed data that was not publically available. This

meant that we had to use the mutation data in a more descriptive way, and had to consider that when computing the scores in the integration. It would be very interesting to perform the statistical analysis of significantly mutated genes. This would allow us to increase the weight assigned to the mutation scores for each gene. It is also worth noticing that all types of mutations (missense mutations, nonsense mutations, etc) have been considered equally in the assignment of a score to a gene. However, it is known that some mutations have a much higher impact in the function of the protein. Hence, considering this might also increase the value of the results.

Bibliography

Alizadeh, A et al. (2000). "Distinct types of diffuse large B-cell lymphoma identified by gene expression profiling". In: *Nature*.

Allison, DB et al. (2006). "Microarray data analysis: from disarray to consolidation and consensus". In: *Nature Reviews*.

Alvarez, Mariano J et al. (2016). "Functional characterization of somatic mutations in cancer using network-based inference of protein activity". In: *Nature*.

Anders, Simon, Paul Theodor Pyl, and Wolfgang Huber (2014). "HTSeq—a Python framework to work with high-throughput sequencing data". In: *Bioinformatics*.

Aryee, MJ et al. (2014). "Minfi: A flexible and comprehensive Bioconductor package for the analysis of Infinium DNA Methylation microarrays." In: *Bioinformatics*.

Barillot et al. (2012). *Computational Systems Biology of Cancer*.

Barretina, J et al. (2012). "The Cancer Cell Line Encyclopedia enables predictive modelling of anticancer drug sensitivity". In: *Nature*.

Bumgarner, R (2013). "Overview of DNA microarrays: types, applications, and their future." In: *Current Protocols in Molecular Biology*.

Carlson, M (2018). *org.Hs.eg.db: Genome wide annotation for Human*.

Cibulskis, Kristian et al. (2013). "Sensitive detection of somatic point mutations in impure and heterogeneous cancer samples". In: *Nature*.

Colaprico, Antonio et al. (2015). "TCGAbiolinks: An R/Bioconductor package for integrative analysis of TCGA data". In: *Nucleic Acids Research*. DOI: 10.1093/nar/gkv1507. URL: http://doi.org/10.1093/nar/gkv1507.

Dees, ND et al. (2012). "MuSiC: Identifying mutational significance in cancer genomes". In: *Genome Research*.

Dudoit, S and J Fridlyand (2002). "A prediction-based resampling method for estimating the number of clusters in a dataset". In: *Genome Biology*.

Duffy, MJ, NC Synnott, and J Crown (2018). "Mutant p53 in breast cancer: potential as a therapeutic target and biomarker". In: *Breast Cancer*.

Durinck, S et al. (2005). "BioMart and Bioconductor: a powerful link between biological databases and microarray data analysis." In: *Bioinformatics*.

Edgar, R, M Domrachev, and AE Lash (2002). "Gene Expression Omnibus: NCBI gene expression and hybridization array data repository". In: *Nucleic Acids Research*.

Ein-Dor, L et al. (2005). "Outcome signature genes in breast cancer: is there a unique set?" In: *Bioinformatics*.

Falcon, S and R Gentleman (2007). "Using GOstats to test gene lists for GO term association." In: *Bioinformatics* 23.2, pp. 257–8.

Finn, RD et al. (2016). "The Pfam protein families database: towards a more sustainable future". In: *Nucleic Acids Research*.

Garcia-Alonso, L et al. (2018). "Transcription Factor Activities Enhance Markers of Drug Sensitivity in Cancer". In: *Cancer Research*.

Gentleman, RC et al. (2004). "Bioconductor: open software development for computational biology and bioinformatics." In: *Genome Biology*.

Goecks, Jeremy et al. (2010). "Galaxy: a comprehensive approach for supporting accessible, reproducible, and transparent computational research in the life sciences". In: *Genome Biology*.

Golub, TR et al. (1999). "Molecular classification of cancer: class discovery and class prediction by gene expression monitoring". In: *Science*.

Hansen, KD et al. (2011). "Increased methylation variation in epigenetic domains across cancer types". In: *Nature*.

Hudson, TJ et al. (2010). "International network of cancer genome projects". In: *Nature*.

Jeffrey, AM and J Wang (2011). "Next-generation transcriptome assembly". In: *Nature*.

Jiao, Y, M Widschwendter, and AE Teschendorff (2014). "A systems-level integrative framework for genome-wide DNA methylation and gene expression data identifies differential gene expression modules under epigenetic control". In: *Bioinformatics*.

Johnstone, Titterington (2009). "Statistical challenges of high-dimensional data". In: *Philosophical Transactions of the Royal Society of London A: Mathematical, Physical and Engineering Sciences*.

Kandula, M et al. (2013). "Phosphatidylinositol 3-kinase (PI3KCA) Oncogene Mutation Analysis and Gene Expression Profiling in Primary Breast Cancer Patients". In: *Asian Pacific Journal of Cancer Prevention*.

Kitano, H. (2002). "Systems Biology: A Brief Overview". In: *Science*.

Kolesnikov, N et al. (2015). "ArrayExpress update—simplifying data submissions". In: *Nucleic Acids Research*.

Lawrence, Michael et al. (2013). "Software for Computing and Annotating Genomic Ranges". In: *PLoS Computational Biology* 9 (8). DOI: 10.1371/journal.pcbi.1003118. URL: http://www.ploscompbiol.org/article/info%3Adoi%2F10.1371%2Fjournal.pcbi.1003118.

Ledford, H (2008). "The death of microarrays?" In: *Nature*.

LoRusso, PM (2016). "Inhibition of the PI3K/AKT/mTOR Pathway in Solid Tumors". In: *Journal of Clinical Oncology*.

Love, MI, W Huber, and S Anders (2014). "Moderated estimation of fold change and dispersion for RNA-seq data with DESeq2". In: *Genome Biology*.

Luo et al. (2013). "Pathview: an R/Bioconductor package for pathway-based data integration and visualization". In: *Bioinformatics* 29.14, pp. 1830–1831. DOI: 10.1093/bioinformatics/btt285.

Martignetti, L et al. (2016). "ROMA: Representation and Quantification of Module Activity from Target Expression Data". In: *Frontiers in Genetics*.

Mayakonda, Anand and Phillip H Koeffler (2016). "Maftools: Efficient analysis, visualization and summarization of MAF files from large-scale cohort based cancer studies." In: *BioRxiv*. DOI: http://dx.doi.org/10.1101/052662.

Mootha, VK et al. (2003). "PGC-1alpha-responsive genes involved in oxidative phosphorylation are coordinately downregulated in human diabetes". In: *Nature Genetics*.

Morganella, Sandro (2010). *GAIA: An R package for genomic analysis of significant chromosomal aberrations*. R package version 2.22.0.

Morganella, Sandro, Stefano Maria Pagnotta, and Michele Ceccarelli (2011). "Finding recurrent copy number alterations preserving within-sample homogeneity". In: *Bioinformatics*.

Pavel, AB, D Sonkin, and A Reddy (2016). "Integrative modeling of multi-omics data to identify cancer drivers and infer patient-specific gene activity". In: *BMC Systems Biology*.

R Core Team (2018). *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing. Vienna, Austria. URL: https://www.R-project.org/.

Ramaswamy, S et al. (2003). "A molecular signature of metastasis in primary solid tumors". In: *Nature Genetics*.

Risso, Davide et al. (2011). "GC-Content Normalization for RNA-Seq Data". In: *BMC Bioinformatics* 12.1, p. 480.

Ritchie, ME et al. (2015). "limma powers differential expression analyses for RNA-sequencing and microarray studies." In: *Nucleic Acids Research*.

Robinson, MD, DJ McCarthy, and GK Smyth (2010). "edgeR: a Bioconductor package for differential expression analysis of digital gene expression data". In: *Bioinformatics*.

Schena, M et al. (1995). "Quantitative Monitoring of Gene Expression Patterns with a Complementary DNA Microarray". In: *Science*.

Symmans, WF et al. (1995). "Breast cancer heterogeneity: evaluation of clonality in primary and metastatic lesions". In: *Human Pathology*.

Tomlins, SA et al. (2005). "Recurrent fusion of TMPRSS2 and ETS transcription factor genes in prostate cancer". In: *Science*.

Tusher, VG, R Tibshirani, and G Chu (2001). "Significance analysis of microarrays applied to the ionizing radiation response". In: *Proceedings of the National Academy of Sciences of the United States of America*.

Vaske, CJ et al. (2010). "Inference of patient-specific pathway activities from multi-dimensional cancer genomics data using PARADIGM". In: *Bioinformatics*.

Veer, LJ van't et al. (2002). "Gene expression profiling predicts clinical outcome of breast cancer". In: *Nature*.

Vijver, MJ van de et al. (2002). "A gene-expression signature as a predictor of survival in breast cancer". In: *The New England Journal of Medicine*.

Vogt, PK (2001). "PI 3-kinase, mTOR, protein synthesis and cancer". In: *Trends in Molecular Medicine*.

Wachter, A and T Beissbarth (2016). "Decoding Cellular Dynamics in Epidermal Growth Factor Signaling Using a New Pathway-Based Integration Approach for Proteomics and Transcriptomics Data". In: *Frontiers in Genetics*.

Wang, Y et al. (2005). "Gene-expression profiles to predict distant metastasis of lymph-node-negative primary breast cancer". In: *The Lancet*.

Weinstein, JN et al. (2013). "The Cancer Genome Atlas Pan-Cancer analysis project". In: *Nature Genetics*.

Wickham, Hadley (2018). *stringr: Simple, Consistent Wrappers for Common String Operations*. R package version 1.3.1. URL: `https://CRAN.R-project.org/package=stringr`.

Appendix A

R Code

A.1 Data Querying

```
setwd("~/Desktop/TFM/")
workdir = getwd()
dataDir = file.path(workdir, "Data/")
robj_dir = file.path(workdir, "Robjects/")

library(TCGAbiolinks)

# Find patients with all the omics

## Find patients with methylation samples for
## both tumor and normal tissue

### Query TCGA

meth_tum = GDCquery("TCGA-BRCA",
                    platform = "Illumina Human Methylation 450",
                    data.category = "DNA Methylation",
                    legacy = F,
                    sample.type = c("Primary solid Tumor"))

meth_normal = GDCquery("TCGA-BRCA",
                       platform = "Illumina Human Methylation 450",
                       data.category = "DNA Methylation",
                       legacy = F,
                       sample.type = c("Solid Tissue Normal"))

### Find intersection between tumor and normal samples

common.methyl = intersect(substr(getResults(meth_tum,
                                  cols = "cases"), 1, 12),
                          substr(getResults(meth_normal,
                                  cols = "cases"), 1, 12))

## Find patients with transcription samples for
## both tumor and normal tissue

### Query TCGA

trans_tum = GDCquery(project = "TCGA-BRCA",
                     data.category = "Transcriptome Profiling",
```

```
                    data.type = "Gene␣Expression␣Quantification",
                     workflow.type = "HTSeq␣-␣Counts",
                     legacy = F,
                     sample.type = c("Primary␣solid␣Tumor"))

trans_normal = GDCquery(project = "TCGA-BRCA",
                        data.category = "Transcriptome␣Profiling",
                        data.type = "Gene␣Expression␣Quantification",
                        workflow.type = "HTSeq␣-␣Counts",
                        legacy = F,
                        sample.type = c("Solid␣Tissue␣Normal"))

### Find intersection between tumor and normal samples

common.trans = intersect(substr(getResults(
                                trans_tum, cols = "cases"),
                                1, 12),
                        substr(getResults(trans_normal,
                                        cols = "cases"),
                                1, 12))

## Find patients with CNV samples for
## both tumor and normal tissue

### Query TCGA

cnv_tum = GDCquery(project = "TCGA-BRCA",
                    data.category = "Copy␣Number␣Variation",
                    legacy = F,
                    sample.type = c("Primary␣solid␣Tumor"),
                   data.type = "Copy␣Number␣Segment")

cnv_normal = GDCquery(project = "TCGA-BRCA",
                    data.category = "Copy␣Number␣Variation",
                    legacy = F,
                    sample.type = c("Solid␣Tissue␣Normal"),
                    data.type = "Copy␣Number␣Segment")

### Find intersection between tumor and normal samples

common.cnv = intersect(substr(getResults(cnv_tum,
                                cols = "cases"),
                            1, 12),
                        substr(getResults(cnv_normal,
                                cols = "cases"),
                        1, 12))

## Find patients with mutation samples for
## both tumor and normal tissue

### Query TCGA

mutations = GDCquery(project = "TCGA-BRCA",
                    experimental.strategy = "WXS",
                    access = "open",
                    data.category = "Simple␣Nucleotide␣Variation",
                    workflow.type =
                        "MuSE␣Variant␣Aggregation␣and␣Masking",
                    legacy = F)
```

```
mutations = strsplit(getResults(mutations, cols = "cases"), ",")
mutations = unlist(mutations)
filtered_mutations = c()

### Find samples of interest

for (i in 1:length(mutations)) {
  if(substr(mutations[i],
            start = 14, stop = 15) == "01" | substr(mutations[i],
                                                    start = 14,
                                                    stop = 15) == "11") {
    filtered_mutations = c(filtered_mutations, mutations[i])
  }
}

filtered_mutations =
        filtered_mutations[duplicated(substr(filtered_mutations,
                                             0, 12))]
filtered_mutations = substr(filtered_mutations, 0, 12)

## Find intersection between all types of data

common.patients = intersect(common.methyl, common.trans)
common.patients = intersect(common.patients, common.cnv)
common.patients = intersect(common.patients, filtered_mutations)

## Save common patients TCGA identifiers as an R object

save(common.patients, file = paste(robj_dir,
    "common_patients.rda", sep = ""))
```

A.2 Data Download

```
setwd("~/Desktop/TFM/")
workdir = getwd()
dataDir = file.path(workdir, "Data/")
robj_dir = file.path(workdir, "Robjects/")
transDir = file.path(dataDir, "Transcription")
methDir = file.path(dataDir, "Methylation", fsep = "")
cnvDir = file.path(dataDir, "CNV", fsep = "")
mutDir = file.path(dataDir, "Mutation")

load(file = file.path(robj_dir, "common_patients.rda", fsep = ""))

library(TCGAbiolinks); library(maftools)

##Data download

###Methylation

meth = GDCquery("TCGA-BRCA",
                barcode = common.patients,
                platform = "Illumina␣Human␣Methylation␣450",
                data.category = "DNA␣Methylation",
                legacy = F,
                sample.type = c("Primary␣solid␣Tumor",
                                "Solid␣Tissue␣Normal"))
```

```
GDCdownload(meth, directory = methDir)

GDCprepare(meth, directory = methDir,
            save = T,
            save.filename = file.path(robj_dir, "meth_data.rda"))

###Transcription

trans = GDCquery(project = "TCGA-BRCA",
                 barcode = common.patients,
                 data.category = "Transcriptome␣Profiling",
                 data.type = "Gene␣Expression␣Quantification",
                 workflow.type = "HTSeq␣-␣Counts", legacy = F,
                 sample.type = c("Primary␣solid␣Tumor",
                                 "Solid␣Tissue␣Normal"))

GDCdownload(trans, directory = transDir)

GDCprepare(trans,
            directory = transDir, save = T,
            save.filename = file.path(robj_dir,
                                      "trans_data.rda"))

###CNVs

cnv = GDCquery(project = "TCGA-BRCA",
                 barcode = common.patients,
                 data.category = "Copy␣Number␣Variation",
                 legacy = F,
                 sample.type = c("Primary␣solid␣Tumor",
                                 "Solid␣Tissue␣Normal"),
                 data.type = "Copy␣Number␣Segment")

GDCdownload(cnv, directory = cnvDir)

GDCprepare(cnv, directory = cnvDir,
            save = T,
            save.filename = file.path(robj_dir,
                                      "cnv_data.rda"))

###Mutations

mutations = GDCquery_Maf(tumor = "BRCA",
                         directory = mutDir,
                         pipelines = "mutect2")
clin = GDCquery_clinic(project = "TCGA-BRCA", type = "Clinical")

colnames(clin)[1] = "Tumor_Sample_Barcode"
clin$Overall_Survival_Status = 1
clin$Overall_Survival_Status[which(clin$vital_status != "dead")] = 0
clin$time = clin$days_to_death
clin$time[is.na(clin$days_to_death)] =
            clin$days_to_last_follow_up[is.na(clin$days_to_death)]

maf = read.maf(maf = mutations,
                 clinicalData = clin, isTCGA = T)

save(maf,file = file.path(robj_dir,
```

```
                    "mut_data.rda"),
                    compress = "xz")
```

A.3 Exploration and outlier filtering

```
setwd("~/Desktop/TFM/")
workdir = getwd()
dataDir = file.path(workdir, "Data/")
robj_dir = file.path(workdir, "Robjects/")

library(SummarizedExperiment)
library(knitr)
library(maftools)

# Transcription

load(file = file.path(robj_dir, "trans_data.rda", fsep = ""))

## Histogram

plot(hist(assay(data[,1]), main = "", breaks = 100), xlab = "Counts",
    cex.axis = 1.5, col = "blue", main = "",
    ylab = "Frequency", cex.lab = 1.5)
plot(hist(log2(assay(data[,1]) + 1), breaks = 100), xlab = "Log2(counts)",
    cex.axis = 1.5, col = "blue", main = "", ylab = "Frequency",
    cex.lab = 1.5)

## PCA

pc = prcomp(t(log2(assay(data) + 1)), scale = FALSE)

bcodes = substr(rownames(pc$x), 1, 15)
tumor = substr(bcodes, 14, 15) == "01"

loads<- round(pc$sdev^2/sum(pc$sdev^2)*100,1)
xlab<-c(paste("PC1",loads[1],"%"))
ylab<-c(paste("PC2",loads[2],"%"))
plot(pc$x[,1], pc$x[,2], xlab = xlab, ylab = ylab, type = "n",
    main = "PC1 vs. PC2", cex.axis = 1.7, cex.lab = 1.3, cex.main = 1.5)
points(pc$x[tumor,1], pc$x[tumor,2], col = "red", pch = 16)
points(pc$x[!tumor,1], pc$x[!tumor,2], col = "blue", pch = 16)
legend("bottomright", legend = c("Tumor", "Normal"), pch = 16, col = c("red", "blue"))

xlab<-c(paste("PC1",loads[1],"%"))
ylab<-c(paste("PC3",loads[3],"%"))
plot(pc$x[,1], pc$x[,3], xlab = xlab, ylab = ylab, type = "n",
    main = "PC1 vs. PC3", cex.axis = 1.7, cex.lab = 1.3)
points(pc$x[tumor,1], pc$x[tumor,3], col = "red", pch = 16)
points(pc$x[!tumor,1], pc$x[!tumor,3], col = "blue", pch = 16)
legend("bottomright", legend = c("Tumor", "Normal"), pch = 16, col = c("red", "blue"))

xlab<-c(paste("PC2",loads[2],"%"))
ylab<-c(paste("PC3",loads[3],"%"))
plot(pc$x[,2], pc$x[,3], xlab = xlab, ylab = ylab, type = "n",
    main = "PC2 vs. PC3", cex.axis = 1.7, cex.lab = 1.3)
points(pc$x[tumor,2], pc$x[tumor,3], col = "red", pch = 16)
points(pc$x[!tumor,2], pc$x[!tumor,3], col = "blue", pch = 16)
legend("bottomright", legend = c("Tumor", "Normal"), pch = 16, col = c("red", "blue"))
```

```
plot(loads, type = "l", ylab = "Amount␣of␣variance", xlab = "Number␣of␣Components",
    main = "Variance␣explained␣vs␣number␣of␣components")

# Outlier identification and removal

out_trans = unique(substr(names(which(pc$x[,2] > 250)), 1, 12))
out_trans_full = names(which(pc$x[,2] > 250))
keep = colnames(assay(data)) %in% out_trans_full

no_out = as.data.frame(assay(data))
no_out = no_out[,!keep]

pc = prcomp(t(log2(no_out + 1)), scale = FALSE)

bcodes = substr(rownames(pc$x), 1, 15)
tumor = substr(bcodes, 14, 15) == "01"

loads<- round(pc$sdev^2/sum(pc$sdev^2)*100,1)
xlab<-c(paste("PC1",loads[1],"%"))
ylab<-c(paste("PC2",loads[2],"%"))
plot(pc$x[,1], pc$x[,2], xlab = xlab, ylab = ylab, type = "n",
    main = "PC1␣vs.␣PC2", cex.axis = 1.7, cex.lab = 1.3, cex.main = 1.5)
points(pc$x[tumor,1], pc$x[tumor,2], col = "red", pch = 16)
points(pc$x[!tumor,1], pc$x[!tumor,2], col = "blue", pch = 16)
legend("bottomright", legend = c("Tumor", "Normal"), pch = 16, col = c("red", "blue"))

xlab<-c(paste("PC1",loads[1],"%"))
ylab<-c(paste("PC3",loads[3],"%"))
plot(pc$x[,1], pc$x[,3], xlab = xlab, ylab = ylab, type = "n",
    main = "PC1␣vs.␣PC3", cex.axis = 1.7, cex.lab = 1.3)
points(pc$x[tumor,1], pc$x[tumor,3], col = "red", pch = 16)
points(pc$x[!tumor,1], pc$x[!tumor,3], col = "blue", pch = 16)
legend("bottomright", legend = c("Tumor", "Normal"), pch = 16, col = c("red", "blue"))

xlab<-c(paste("PC2",loads[2],"%"))
ylab<-c(paste("PC3",loads[3],"%"))
plot(pc$x[,2], pc$x[,3], xlab = xlab, ylab = ylab, type = "n",
    main = "PC2␣vs.␣PC3", cex.axis = 1.7, cex.lab = 1.3)
points(pc$x[tumor,2], pc$x[tumor,3], col = "red", pch = 16)
points(pc$x[!tumor,2], pc$x[!tumor,3], col = "blue", pch = 16)
legend("bottomright", legend = c("Tumor", "Normal"), pch = 16, col = c("red", "blue"))

plot(loads, type = "l", ylab = "Amount␣of␣variance", xlab = "Number␣of␣Components",
    main = "Variance␣explained␣vs␣number␣of␣components")

# Update file structure

dataDir_noout = file.path(workdir, "Data_no_out/")
methDir2 = file.path(dataDir_noout, "Methylation", fsep = "")
transDir2 = file.path(dataDir_noout, "Transcription", fsep = "")
cnvDir2 = file.path(dataDir_noout, "CNV", fsep = "")
mutDir2 = file.path(dataDir_noout, "Mutations", fsep = "")

load(file = file.path(robj_dir, "common_patients.rda", fsep = ""))

common.patients = common.patients[!common.patients %in% out_trans]
save(common.patients, file = file.path(robj_dir, "common_patients.rda"))
```

```
## Methylation

meth = GDCquery("TCGA-BRCA", barcode = common.patients,
                platform = "Illumina␣Human␣Methylation␣450",
                data.category = "DNA␣Methylation",
                legacy = F,
                sample.type = c("Primary␣solid␣Tumor", "Solid␣Tissue␣Normal"))

GDCdownload(meth, directory = methDir2)

GDCprepare(meth, directory = methDir2, save = T,
           save.filename = file.path(robj_dir, "meth_data_no_out.rda"))

## Transcription

trans = GDCquery(project = "TCGA-BRCA", barcode = common.patients,
                data.category = "Transcriptome␣Profiling",
                data.type = "Gene␣Expression␣Quantification",
                workflow.type = "HTSeq␣-␣Counts", legacy = F,
                sample.type = c("Primary␣solid␣Tumor", "Solid␣Tissue␣Normal"))

GDCdownload(trans, directory = transDir2)

GDCprepare(trans, directory = transDir2, save = T,
           save.filename = file.path(robj_dir, "trans_data_no_out.rda"))

## CNV

cnv = GDCquery(project = "TCGA-BRCA", barcode = common.patients,
                data.category = "Copy␣Number␣Variation", legacy = F,
                sample.type = c("Primary␣solid␣Tumor", "Solid␣Tissue␣Normal"),
                data.type = "Copy␣Number␣Segment")

GDCdownload(cnv, directory = cnvDir2)

GDCprepare(cnv, directory = cnvDir2, save = T,
           save.filename = file.path(robj_dir, "cnv_data_no_out.rda"))

## Mutations

mutations = GDCquery_Maf(tumor = "BRCA", directory = mutDir2, pipelines = "mutect2")
clin = GDCquery_clinic(project = "TCGA-BRCA", type = "Clinical")

# Methylation

load(file = file.path(robj_dir, "meth_data_no_out.rda", fsep = ""))

methylation = as.data.frame(assay(data))

table(is.na(methylation))
methylation = na.omit(methylation)

## PCA

pc = prcomp(t(methylation), scale = FALSE)

bcodes = substr(rownames(pc$x), 1, 15)
tumor = substr(bcodes, 14, 15) == "01"
```

```
loads<- round(pc$sdev^2/sum(pc$sdev^2)*100,1)
xlab<-c(paste("PC1",loads[1],"%"))
ylab<-c(paste("PC2",loads[2],"%"))
plot(pc$x[,1], pc$x[,2], xlab = xlab, ylab = ylab, type = "n",
    main = "PC1 vs. PC2", cex.axis = 1.5, cex.lab = 1.3, cex.main = 1.5)
points(pc$x[tumor,1], pc$x[tumor,2], col = "red", pch = 16)
points(pc$x[!tumor,1], pc$x[!tumor,2], col = "blue", pch = 16)
legend("topright", legend = c("Tumor", "Normal"), pch = 16, col = c("red", "blue"))

xlab<-c(paste("PC1",loads[1],"%"))
ylab<-c(paste("PC3",loads[3],"%"))
plot(pc$x[,1], pc$x[,3], xlab = xlab, ylab = ylab, type = "n",
    main = "PC1 vs. PC3", cex.axis = 1.5, cex.lab = 1.3, cex.main = 1.5)
points(pc$x[tumor,1], pc$x[tumor,3], col = "red", pch = 16)
points(pc$x[!tumor,1], pc$x[!tumor,3], col = "blue", pch = 16)
legend("topright", legend = c("Tumor", "Normal"), pch = 16, col = c("red", "blue"))

xlab<-c(paste("PC2",loads[2],"%"))
ylab<-c(paste("PC3",loads[3],"%"))
plot(pc$x[,2], pc$x[,3], xlab = xlab, ylab = ylab, type = "n",
    main = "PC2 vs. PC3", cex.axis = 1.5, cex.lab = 1.3, cex.main = 1.5)
points(pc$x[tumor,2], pc$x[tumor,3], col = "red", pch = 16)
points(pc$x[!tumor,2], pc$x[!tumor,3], col = "blue", pch = 16)
legend("topright", legend = c("Tumor", "Normal"), pch = 16, col = c("red", "blue"))

plot(loads, type = "l", ylab = "Amount of variance", xlab = "Number of Components",
    main = "Variance explained vs number of components")

## B-val density

set.seed(444)
rand_smp = sample(rownames(methylation), 10000, replace = F)
bcodes = substr(colnames(methylation), 1, 15)
tumor = substr(bcodes, 14, 15) == "01"

plot(density(methylation[rand_smp, 1]), type = "n", ylim = c(0,3),
    main = "Density of B-values for tumor and normal samples", cex.axis = 1.5, xlab = "",
    ylab = "", cex.main = 1.5)
for (i in 1:length(bcodes)) {
  if (tumor[i] == T) {
    points(density(methylation[rand_smp, i]), col = "red", type = "l")
  }
  else {
    points(density(methylation[rand_smp, i]), col = "blue", type = "l")
  }
}
legend("topright", legend = c("Tumor", "Normal"), col = c("red", "blue"), lty = 1)

# CNV

load(file = file.path(robj_dir, "cnv_data_no_out.rda", fsep = ""))

## Segment Meant density

bcodes = substr(unique(data$Sample),1,15)
bcodes_full = unique(data$Sample)
tumor = substr(bcodes, 14, 15) == "01"

plot(density(subset(data$Segment_Mean, data$Sample == bcodes_full[1])),
```

```
    type = "n", main = "Density␣of␣Segment␣Mean␣for␣tumor␣and␣normal␣samples",
    ylim = c(0,5), xlab = "", ylab = "", cex.axis = 1.5, cex.main = 1.5)
for (i in 1:length(bcodes_full)) {
  if (tumor[i] == T) {
    points(density(subset(data$Segment_Mean, data$Sample == bcodes_full[i])),
           col = "red", type = "l")
  }
  else {
    points(density(subset(data$Segment_Mean, data$Sample == bcodes_full[i])),
           col = "blue", type = "l")
  }
}
legend("topright", legend = c("Tumor", "Normal"), col = c("red", "blue"), lty = 1)

plot(density(subset(data$Segment_Mean, data$Sample == bcodes_full[i])), type = "n",
    main = "Density␣of␣Segment␣Mean␣for␣tumor␣samples",
    ylim = c(0,5))
for (i in 1:length(bcodes_full)) {
  if (tumor[i] == T) {
    points(density(subset(data$Segment_Mean, data$Sample == bcodes_full[i])),
    col = "red", type = "l")
  }
}

plot(density(subset(data$Segment_Mean, data$Sample == bcodes_full[i])), type = "n",
    main = "Density␣of␣Segment␣Mean␣for␣normal␣samples",
    ylim = c(0,5))
for (i in 1:length(bcodes_full)) {
  if (tumor[i] == F) {
    points(density(subset(data$Segment_Mean, data$Sample == bcodes_full[i])),
           col = "blue", type = "l")
  }
}

# Mutations

load(file = file.path(robj_dir, "mut_data.rda", fsep = ""))

maf2 = subsetMaf(maf, tsb = common.patients, mafObj = TRUE)

table(maf2@data$IMPACT)

kable(table(mutations_filtered$IMPACT))
kable(table(mutations_filtered$One_Consequence))
```

A.4 Differential expression analysis, functional analysis and visualization

```
setwd("~/Desktop/TFM/")
workdir = getwd()
dataDir = file.path(workdir, "Data_no_out/")
transDir = file.path(dataDir, "Transcription/", fsep = "")
robj_dir = file.path(workdir, "Robjects/")

library(TCGAbiolinks); library(biomaRt); library(annotate); library(org.Hs.eg.db)
library(GOstats); library(clusterProfiler); library(pathview)

# Load data
```

```
load(file = file.path(robj_dir, "trans_data_no_out.rda"))
load(file.path(robj_dir, "common_patients.rda", fsep = ""))

# Preprocessing

dataPrep = TCGAanalyze_Preprocessing(object = data, cor.cut = 0.6,
                                     datatype = "HTSeq - Counts")

# Normalization

dataNorm = TCGAanalyze_Normalization(tabDF = dataPrep, geneInfo = geneInfoHT,
                                     method = "gcContent")

par(mfrow=c(2,1))
boxplot(dataPrep, outline = FALSE, names = FALSE, main = "Before Normalization")
boxplot(dataNorm, outline = FALSE, names = FALSE, main = "After Normalization")
dev.off()

# Filtering

dataFilt = TCGAanalyze_Filtering(tabDF = dataNorm, method = "quantile", qnt.cut =
0.25)

# DEA

query = GDCquery(project = "TCGA-BRCA", barcode = common.patients,
                 data.category = "Transcriptome Profiling",
                 data.type = "Gene Expression Quantification",
                 workflow.type = "HTSeq - Counts",
                 sample.type = c("Primary solid Tumor", "Solid Tissue Normal"))

bcodes = getResults(query, cols = "cases")

trans_tumor = TCGAquery_SampleTypes(barcode = bcodes, typesample = "TP")
trans_normal = TCGAquery_SampleTypes(barcode = bcodes, typesample = "NT")

dataDEGs = TCGAanalyze_DEA(mat1 = dataFilt[,trans_normal], mat2 = dataFilt[,trans_tumor],
                           Cond1type = "Normal", Cond2type = "Tumor", fdr.cut = 0.001,
                           logFC.cut = 1,
                           method = "glmLRT")

# Volcano plot

plot(dataDEGs$logFC, -log10(dataDEGs$PValue), xlab = "Fold Change",
    ylab = "-log10(p-val)", main = "Volcano plot", col = "brown")

# Functional analysis

# GOstats

genes = rownames(dataFilt)

mart = useMart("ensembl",dataset="hsapiens_gene_ensembl")
annot = getBM(attributes = c("hgnc_symbol", "ensembl_gene_id"),
            filters = "ensembl_gene_id", values = genes, mart = mart)

table(annot$hgnc_symbol == "")
keep = annot$hgnc_symbol != ""
```

```
annot = annot[keep,]

table(duplicated(annot$ensembl_gene_id))
annot = annot[!duplicated(annot$ensembl_gene_id),]

entrezUniverse = annot$entrezgene
entrezUniverse = entrezUniverse[!duplicated(entrezUniverse)]

geneIds = dataDEGs$entrezgene
geneIds = geneIds[!duplicated(geneIds)]

GOparams = new("GOHyperGParams", geneIds = geneIds,
               universeGeneIds = entrezUniverse, annotation = "org.Hs.eg.db",
               ontology = "BP", pvalueCutoff = 0.001, conditional = FALSE,
               testDirection = "over")

GOhyper = hyperGTest(GOparams)

GOfilename = file.path(workdir, "Trans_GOResults.html")

htmlReport(GOhyper, file = GOfilename, summary.args = list("htmlLinks" = TRUE))

# Save results

save(dataDEGs, file = file.path(robj_dir, "trans_results.rda"))

# Pathview

GenelistComplete = rownames(dataFilt)

# DEGs TopTable
dataDEGsFiltLevel = TCGAanalyze_LevelTab(dataDEGs,"Normal","Tumor",
                                         dataFilt[,trans_normal],
                                         dataFilt[,trans_tumor])

dataDEGsFiltLevel$GeneID = 0


# Converting Gene symbol to geneID
eg = as.data.frame(bitr(dataDEGsFiltLevel$external_gene_name,
                        fromType="SYMBOL",
                        toType="ENTREZID",
                        OrgDb="org.Hs.eg.db"))
eg = eg[!duplicated(eg$SYMBOL),]

dataDEGsFiltLevel = dataDEGsFiltLevel[dataDEGsFiltLevel$external_gene_name %in% eg$SYMBOL,]

dataDEGsFiltLevel = dataDEGsFiltLevel[order(dataDEGsFiltLevel$external_gene_name,decreasing=F
eg = eg[order(eg$SYMBOL,decreasing=FALSE),]

all(eg$SYMBOL == dataDEGsFiltLevel$external_gene_name)

dataDEGsFiltLevel$GeneID = eg$ENTREZID

dataDEGsFiltLevel_sub = subset(dataDEGsFiltLevel, select = c("entrezgene", "logFC"))
genelistDEGs = as.numeric(dataDEGsFiltLevel_sub$logFC)
names(genelistDEGs) = dataDEGsFiltLevel_sub$entrezgene

hsa04110 = pathview::pathview(gene.data  = genelistDEGs,
```

```
                              pathway.id = "hsa04110",
                              species    = "hsa",
                              limit = list(gene=as.integer(max(abs(genelistDEGs)))))

hsa04010 = pathview::pathview(gene.data  = genelistDEGs,
                              pathway.id = "hsa04010",
                              species    = "hsa",
                              limit = list(gene=as.integer(max(abs(genelistDEGs)))))

hsa04115 = pathview::pathview(gene.data  = genelistDEGs,
                              pathway.id = "hsa04115",
                              species    = "hsa",
                              limit = list(gene=as.integer(max(abs(genelistDEGs)))))

hsa04151 = pathview::pathview(gene.data  = genelistDEGs,
                              pathway.id = "hsa04151",
                              species    = "hsa",
                              limit = list(gene=as.integer(max(abs(genelistDEGs)))))

hsa05224 = pathview::pathview(gene.data  = genelistDEGs,
                             pathway.id = "hsa05224",
                             species    = "hsa",
                             limit = list(gene=as.integer(max(abs(genelistDEGs)))))
```

A.5 Differential methylation analysis, functional analysis and visualization

```
setwd("~/Desktop/TFM/")
workdir = getwd()
dataDir = file.path(workdir, "Data_no_out/")
methDir = file.path(dataDir, "Methylation/", fsep = "")
robj_dir = file.path(workdir, "Robjects/")

library(TCGAbiolinks)
library(SummarizedExperiment)
library(stringr)
library(GOstats)
library(biomaRt)

# Load data

load(file = file.path(robj_dir, "meth_data_no_out.rda"))

# Preprocess

met = subset(data, rowSums(assay(data)) != 0)

# Mean methylation

TCGAvisualize_meanMethylation(met, groupCol = "definition", group.legend  = "Groups",
                              filename = "meanMethyl.pdf", print.pvalue = TRUE)

# Differential Methylation Analysis

met = TCGAanalyze_DMR(met,groupCol = "definition",
                      group1 = "Solid␣Tissue␣Normal", group2 = "Primary␣solid␣Tumor",
                      p.cut = 10^-3, diffmean.cut = 0.35, save = FALSE, legend = "State",
                      plot.filename = "Meth.png", overwrite = T)
```

```
# Result filtering

met_annot = subset(as.data.frame(met@rowRanges), as.data.frame(met@rowRanges$Gene_Symbol) !=
met_annot_sig = subset(met_annot,
                 met_annot$status.Solid.Tissue.Normal.Primary.solid.Tumor != "Not Significant"

# Process to get single rows

#Subset the whole table to get the relevant data
met_annot_short = met_annot_sig[, c(1,2,6,7,8,10,11,12,20,21)]

#Find if there are semicolons in the Gene_Symbol field and count them for each row in the tab
semicolons = replace(str_locate(met_annot_short$Gene_Symbol, ";")[,1],
                     is.na(str_locate(met_annot_short$Gene_Symbol, ";")[,1]), 0)
int_df = data.frame()
final_df = data.frame(matrix(ncol = 10, nrow = 0))
colnames(final_df) = colnames(met_annot_short)

#Loop in each row in the data table
for (i in 1:nrow(met_annot_short)) {
  #If there is one or more semicolons in that row's Gene Symbol
  if (semicolons[i] > 0){
    #Split the values by semicolon
    split_sym = unlist(strsplit(met_annot_short$Gene_Symbol[i], ";"))
    split_type = unlist(strsplit(met_annot_short$Gene_Type[i], ";"))
    split_tID = unlist(strsplit(as.character(met_annot_short$Transcript_ID[i]), ";"))
    split_PosTSS = unlist(strsplit(met_annot_short$Position_to_TSS[i], ";"))

    #Save the split data in an intermediary dataframe
    int_df = data.frame(seqnames = met_annot_short$seqnames[i], start = met_annot_short$start
                        Composite.Element.REF = met_annot_short$Composite.Element.REF[i],
                        Gene_Symbol = split_sym, Gene_Type = split_type, Position_to_TSS = sp
                        CGI_Coordinate = met_annot_short$CGI_Coordinate[i],
                        Feature_Type = met_annot_short$Feature_Type[i],
                        p.value.adj.Solid.Tissue.Normal.Primary.solid.Tumor = met_annot_short
                        status.Solid.Tissue.Normal.Primary.solid.Tumor = met_annot_short$stat
    final_df = rbind(final_df, int_df)
  }
  #If the row contains only one gene or transcript
  else {
    int_df = met_annot_short[i,]
    final_df = rbind(final_df, int_df)
  }
  print(i)
}

# Recover probes mapping within 200 bp of protein coding genes

final_df = subset(final_df,
                  abs(as.numeric(as.character(final_df$Position_to_TSS))) < 201
                                          & final_df$Gene_Type == "protein_coding")

# Save results

save(final_df, file = file.path(robj_dir, "meth_results.rda"))

# Get Probe closest to TSS for each gene
```

```
un_gene = unique(final_df$Gene_Symbol)
final_df$Position_to_TSS = as.numeric(as.character(final_df$Position_to_TSS))
int_df = data.frame()
unique_df = data.frame()
for (gene in 1:length(un_gene)) {
  if (gene == 1) {
    gene_TSSs = subset(final_df, final_df$Gene_Symbol == un_gene[gene])
    unique_df = subset(gene_TSSs, abs(gene_TSSs$Position_to_TSS) == min(abs(gene_TSSs$Positio
  }
  else {
    gene_TSSs = subset(final_df, final_df$Gene_Symbol == un_gene[gene])
    int_df = subset(gene_TSSs, abs(gene_TSSs$Position_to_TSS) == min(abs(gene_TSSs$Position_t
    unique_df = rbind(unique_df, int_df)
  }
}

unique_df = unique_df[!duplicated(unique_df),]

save(unique_df, file = file.path(robj_dir, "meth_results_unique.rda"))

# GO and KEGG enrichment analysis

## Get all gene SYMBOLS in methylation chip

met = subset(data, rowSums(assay(data)) != 0)
met = as.data.frame(rowRanges(met))
met = subset(met, met$Gene_Symbol != ".")

semicolons = replace(str_locate(met$Gene_Symbol, ";")[,1], is.na(str_locate(met$Gene_Symbol,
genes = c()
for (i in 1:length(met$Gene_Symbol)) {
  #If there is one or more semicolons in that row's Gene Symbol
  if (semicolons[i] > 0){
    #Split the values by semicolon
    genes = c(genes, unlist(strsplit(met$Gene_Symbol[i], ";")))
  }
  #If the row contains only one gene or transcript
  else {
    genes = c(genes, met$Gene_Symbol[i])
  }
  print(i)
}

## Due to the computation time it gets to get the genes object, let's save it
save(genes, file = file.path(robj_dir, "meth_gene_symbols_all.rda"))

Universe = unique(genes)

## Annotate all genes with entrez ID

mart = useMart("ensembl",dataset="hsapiens_gene_ensembl")
annot = getBM(attributes = c("hgnc_symbol", "ensembl_gene_id", "entrezgene"),
              filters = "hgnc_symbol", values = Universe, mart = mart)
entrezUniverse = unique(annot$entrezgene)

## Get Differentially Methylated Gene Symbols

geneIds = as.character(unique(unique_df$Gene_Symbol))
```

```
## Annotate all Differentially Methylated Gene with entrez ID

annot_DMGs = getBM(attributes = c("hgnc_symbol", "ensembl_gene_id", "entrezgene"),
                   filters = "hgnc_symbol", values = geneIds, mart = mart)
annot_DMGs = annot_DMGs[!duplicated(annot_DMGs$entrezgene),]
entrez_DMGs = annot_DMGs$entrezgene

## Set parameters for GO enrichment analysis and run it

GOparams = new("GOHyperGParams", geneIds = entrez_DMGs,
               universeGeneIds = entrezUniverse, annotation = "org.Hs.eg.db",
               ontology = "BP", pvalueCutoff = 0.001, conditional = FALSE,
               testDirection = "over")

GOhyper = hyperGTest(GOparams)

GOfilename = file.path(workdir, "Meth_GOResults.html")

htmlReport(GOhyper, file = GOfilename, summary.args = list("htmlLinks" = TRUE))

# Pathview

## Annotate all DMG Symbols with an Entrez ID

eg = as.data.frame(bitr(unique_df$Gene_Symbol, fromType = "SYMBOL",
                   toType = "ENTREZID", OrgDb = "org.Hs.eg.db"))

## Get results for annotated DMGs
Genes = unique_df[unique_df$Gene_Symbol %in% eg$SYMBOL, ]
Genes = Genes[order(Genes$Gene_Symbol, decreasing = F),]

eg = eg[order(eg$SYMBOL),]

Genes$ENTREZID = eg$ENTREZID

## Subset the data we need
Genes_sub = Genes[,10:11]
status = Genes_sub$status.Solid.Tissue.Normal.Primary.solid.Tumor
status = as.character(status)

## Convert Hypermethylated -> 1 and Hypomethylated to -1 for the colour scale
status = replace(status, status == "Hypermethylated", "-1")
status = replace(status, status == "Hypomethylated", "1")
Genes_sub$status.Solid.Tissue.Normal.Primary.solid.Tumor = as.numeric(status)

## Create a named vector containing the DMGs and their state
genelistDMGs = as.numeric(Genes_sub$status.Solid.Tissue.Normal.Primary.solid.Tumor)
names(genelistDMGs) = Genes_sub$ENTREZID

## Get the maps

hsa04110 = pathview::pathview(gene.data  = genelistDMGs,
                              pathway.id = "hsa04110",
                              species    = "hsa",
                              limit = list(gene=as.integer(max(abs(genelistDMGs)))))

hsa04010 = pathview::pathview(gene.data  = genelistDMGs,
                              pathway.id = "hsa04010",
                              species    = "hsa",
```

```
                              limit = list(gene=as.integer(max(abs(genelistDMGs)))))

hsa04115 = pathview::pathview(gene.data  = genelistDMGs,
                              pathway.id = "hsa04115",
                              species    = "hsa",
                              limit = list(gene=as.integer(max(abs(genelistDMGs)))))

hsa04151 = pathview::pathview(gene.data  = genelistDMGs,
                              pathway.id = "hsa04151",
                              species    = "hsa",
                              limit = list(gene=as.integer(max(abs(genelistDMGs)))))

hsa05224 = pathview::pathview(gene.data  = genelistDMGs,
                              pathway.id = "hsa05224",
                              species    = "hsa",
                              limit = list(gene=as.integer(max(abs(genelistDMGs)))))
```

A.6 Copy Number Variation recurrency analysis, functional analysis and visualization

```
setwd("~/Desktop/TFM/")
workdir = getwd()
dataDir = file.path(workdir, "Data_no_out/")
cnvDir = file.path(dataDir, "CNV/", fsep = "")
robj_dir = file.path(workdir, "Robjects/")

library(TCGAbiolinks)
library(SummarizedExperiment)
library(stringr)
library(gaia)
library(GenomicRanges)

# Load data

load(file = file.path(robj_dir, "cnv_data_no_out.rda"))

# Load marker matrix

markersMatrix = read.table("~/Desktop/TFM/Data/CNV/snp6.na35.liftoverhg38.txt",
                           sep = "\t", header = T)

# Format cnv matrix

cnvMatrix = cbind(data,Label=NA)
cnvMatrix[cnvMatrix[,"Segment_Mean"] < -0.4,"Label"] = 0
cnvMatrix[cnvMatrix[,"Segment_Mean"] > 0.4,"Label"] = 1
cnvMatrix = cnvMatrix[!is.na(cnvMatrix$Label),]
cnvMatrix = cnvMatrix[,-6]
colnames(cnvMatrix) = c("Sample.Name", "Chromosome", "Start", "End",
                        "Num.of.Markers", "Aberration")
cnvMatrix[cnvMatrix$Chromosome == "X","Chromosome"] = 23
cnvMatrix[cnvMatrix$Chromosome == "Y","Chromosome"] = 24
cnvMatrix$Chromosome = as.integer(cnvMatrix$Chromosome)

# Format markers matrix

colnames(markersMatrix)[1:3] = c("Probe.Name", "Chromosome", "Start")
markersMatrix$Chromosome = as.character(markersMatrix$Chromosome)
```

```
markersMatrix[markersMatrix$Chromosome == "X","Chromosome"] = 23
markersMatrix[markersMatrix$Chromosome == "Y","Chromosome"] = 24
markersMatrix$Chromosome = as.integer(markersMatrix$Chromosome)
markerID = paste(markersMatrix$Chromosome,markersMatrix$Start, sep = ":")
markersMatrix = markersMatrix[!duplicated(markerID),]

# Load data into r objects for Gaia

markers_obj = load_markers(markersMatrix)
n = length(unique(cnvMatrix$Sample))
cnv_obj = load_cnv(cnvMatrix, markers_obj, n)

# Run Gaia

set.seed(444)
results = runGAIA(cnv_obj, markers_obj, output_file_name = "GAIA_BRCA.txt",
                  aberrations = -1,  chromosomes = -1,  approximation = TRUE,
                  num_iterations = 1000000, threshold = 0.0001, hom_threshold = 0.12)

# Create table with recurrent CNVs

RecCNV = t(apply(results,1,as.numeric))
colnames(RecCNV) = colnames(results)
RecCNV = as.data.frame(RecCNV)
RecCNV = cbind(RecCNV, score = 0)
minval = format(min(RecCNV[RecCNV[,"q-value"] != 0,"q-value"]), scientific = FALSE)
minval = substring(minval,1, nchar(minval) - 1)
RecCNV[RecCNV[,"q-value"] == 0,"q-value"] = as.numeric(minval)
RecCNV[,"score"] = sapply(RecCNV[,"q-value"],function(x) -log10(as.numeric(x)))
head(RecCNV[RecCNV[,"q-value"] == as.numeric(minval),])

# Plot results

threshold = 0.3
gaiaCNVplot(RecCNV,threshold)

# Annotation

genes = TCGAbiolinks:::get.GRCh.bioMart(genome = "hg38")
genes = genes[genes$external_gene_name != "" & genes$chromosome_name %in% c(1:22,"X","Y"),]
genes[genes$chromosome_name == "X", "chromosome_name"] = 23
genes[genes$chromosome_name == "Y", "chromosome_name"] = 24
genes$chromosome_name = sapply(genes$chromosome_name,as.integer)
genes = genes[order(genes$start_position),]
genes = genes[order(genes$chromosome_name),]
genes = genes[,c("external_gene_name", "chromosome_name", "start_position","end_position")]
colnames(genes) = c("GeneSymbol","Chr","Start","End")
genes_GR = makeGRangesFromDataFrame(genes,keep.extra.columns = TRUE)

sCNV = RecCNV[RecCNV[,"q-value"] <= threshold,c(1:4,6)] #Revisar
sCNV = sCNV[order(sCNV[,3]),]
sCNV = sCNV[order(sCNV[,1]),]
colnames(sCNV) = c("Chr","Aberration","Start","End","q-value")
sCNV = as.data.frame(sCNV)

remove = which((sCNV$End - sCNV$Start) < 0)
sCNV = sCNV[-remove, ]

sCNV_GR = makeGRangesFromDataFrame(sCNV,keep.extra.columns = TRUE)
```

```
hits = findOverlaps(genes_GR, sCNV_GR, type = "within")
sCNV_ann = cbind(sCNV[subjectHits(hits),],genes[queryHits(hits),])
AberrantRegion = paste0(sCNV_ann[,1],":",sCNV_ann[,3],"-",sCNV_ann[,4])
GeneRegion = paste0(sCNV_ann[,7],":",sCNV_ann[,8],"-",sCNV_ann[,9])
AmpDel_genes = cbind(sCNV_ann[,c(6,2,5)],AberrantRegion,GeneRegion)
AmpDel_genes[AmpDel_genes[,2] == 0,2] = "Del"
AmpDel_genes[AmpDel_genes[,2] == 1,2] = "Amp"
rownames(AmpDel_genes) = NULL

# Save data

save(AmpDel_genes, file = file.path(robj_dir, "cnv_results.rda"))

# GO Enrichment analysis

## Get geme annottions

genes = TCGAbiolinks:::get.GRCh.bioMart(genome = "hg38")
genes = genes[genes$external_gene_name != "" & genes$chromosome_name %in% c(1:22,"X","Y"),]
genes[genes$chromosome_name == "X", "chromosome_name"] = 23
genes[genes$chromosome_name == "Y", "chromosome_name"] = 24
genes$chromosome_name = sapply(genes$chromosome_name,as.integer)
genes = genes[order(genes$start_position),]
genes = genes[order(genes$chromosome_name),]
genes = genes[,c("external_gene_name", "chromosome_name", "start_position","end_position")]
colnames(genes) = c("GeneSymbol","Chr","Start","End")
genes_GR = makeGRangesFromDataFrame(genes,keep.extra.columns = TRUE)

allregions = data[,2:4]
allregions = allregions[order(allregions$Chromosome),]
allregions = allregions[order(allregions$Start),]
allregions[allregions$Chromosome == "X", "Chromosome"] = 23
allregions[allregions$Chromosome == "Y", "Chromosome"] = 24
allregions_GR = makeGRangesFromDataFrame(allregions,keep.extra.columns = TRUE)

hits = findOverlaps(genes_GR, allregions_GR, type = "within")
allregions_ann = cbind(allregions[subjectHits(hits),],genes[queryHits(hits),])

allgenes = unique(allregions_ann$GeneSymbol)

mart = useMart("ensembl",dataset="hsapiens_gene_ensembl")
annot = getBM(attributes = c("hgnc_symbol", "ensembl_gene_id", "entrezgene"),
              filters = "hgnc_symbol", values = genes, mart = mart)
entrezUniverse = annot[!duplicated(annot$entrezgene),]
entrezUniverse = entrezUniverse$entrezgene

geneIds = AmpDel_genes$GeneSymbol
annot_ampdel = getBM(attributes = c("hgnc_symbol", "ensembl_gene_id", "entrezgene"),
                     filters = "hgnc_symbol", values = geneIds, mart = mart)
annot_ampdel = annot_ampdel[!duplicated(annot_ampdel$entrezgene),]
entrez_ampdel = annot_ampdel$entrezgene

GOparams = new("GOHyperGParams", geneIds = entrez_ampdel,
               universeGeneIds = entrezUniverse, annotation = "org.Hs.eg.db",
               ontology = "BP", pvalueCutoff = 0.001, conditional = FALSE,
               testDirection = "over")

GOhyper = hyperGTest(GOparams)
```

```
GOfilename = file.path(workdir, "CNV_GOResults.html")

htmlReport(GOhyper, file = GOfilename, summary.args = list("htmlLinks" = TRUE))

# Pathview

eg = as.data.frame(bitr(AmpDel_genes$GeneSymbol, fromType = "SYMBOL",
                        toType = "ENTREZID", OrgDb = "org.Hs.eg.db"))
eg = eg[!duplicated(eg$SYMBOL),]

## Get the results for the annotated amp_del genes
Genes = AmpDel_genes[AmpDel_genes$GeneSymbol %in% eg$SYMBOL, ]
Genes = Genes[!duplicated(Genes$GeneSymbol),]
Genes = Genes[order(Genes$GeneSymbol, decreasing = F),]

eg = eg[order(eg$SYMBOL),]

Genes$ENTREZID = eg$ENTREZID

## Subset the data we need
Genes_sub = Genes[,c(2,6)]
status = Genes_sub$Aberration
status = as.character(status)
status = replace(status, status == "Amp", "1")
status = replace(status, status == "Del", "-1")
status = as.numeric(status)
Genes_sub$Aberration = status

ampdelList = Genes_sub$Aberration
names(ampdelList) = Genes_sub$ENTREZID

## Get the maps

hsa04110 = pathview::pathview(gene.data  = ampdelList,
                              pathway.id = "hsa04110",
                              species    = "hsa",
                              limit = list(gene=as.integer(max(abs(ampdelList)))))

hsa04010 = pathview::pathview(gene.data  = ampdelList,
                              pathway.id = "hsa04010",
                              species    = "hsa",
                              limit = list(gene=as.integer(max(abs(ampdelList)))))

hsa04115 = pathview::pathview(gene.data  = ampdelList,
                              pathway.id = "hsa04115",
                              species    = "hsa",
                              limit = list(gene=as.integer(max(abs(ampdelList)))))

hsa04151 = pathview::pathview(gene.data  = ampdelList,
                              pathway.id = "hsa04151",
                              species    = "hsa",
                              limit = list(gene=as.integer(max(abs(ampdelList)))))

hsa05224 = pathview::pathview(gene.data  = ampdelList,
                              pathway.id = "hsa05224",
                              species    = "hsa",
                              limit = list(gene=as.integer(max(abs(ampdelList)))))
```

A.7 DNA-seq exploration and Analysis

```
setwd("~/Desktop/TFM/")
workdir = getwd()
dataDir = file.path(workdir, "Data/")
mutDir = file.path(dataDir, "Mutation/", fsep = "")
robj_dir = file.path(workdir, "Robjects/")

library(TCGAbiolinks)
library(SummarizedExperiment)
library(stringr)
library(gaia)
library(GenomicRanges)
library(maftools)

# Load data

load(file = file.path(robj_dir, "mut_data.rda"))
load(file = file.path(robj_dir, "common_patients.rda"))
maf2 = subsetMaf(maf, tsb = common.patients, mafObj = TRUE)

# MAF Summary

plotmafSummary(maf = maf2, rmOutlier = TRUE, addStat = 'median', dashboard = TRUE)

# Oncoplot

oncoplot(maf = maf2, top = 20, legendFontSize = 8)

# Lollipop plots

lollipopPlot(maf2, "TP53")
lollipopPlot(maf2, "PIK3CA")

## Analysis

geneMat = maf2@gene.summary
geneMat = geneMat[,c(1,11)]
geneMat$Prop = geneMat$total/length(common.patients)
geneMat = geneMat[order(geneMat$Prop, decreasing = T),]
geneMat$Score = abs(1/log10(geneMat$Prop))

save(geneMat, file = file.path(robj_dir, "mut_results.rda"))

# Pathview

eg = as.data.frame(bitr(geneMat$Hugo_Symbol, fromType = "SYMBOL", toType = "ENTREZID", OrgDb
eg = eg[!duplicated(eg$SYMBOL),]

## Get the results for the annotated amp_del genes
Genes = geneMat[geneMat$Hugo_Symbol %in% eg$SYMBOL, ]
Genes = Genes[!duplicated(Genes$Hugo_Symbol),]
Genes = Genes[order(Genes$Hugo_Symbol, decreasing = F),]

eg = eg[order(eg$SYMBOL),]

Genes$ENTREZID = eg$ENTREZID
```

```
mutList = abs(Genes$Score)
names(mutList) = Genes$ENTREZID

## Get the maps

hsa04110 = pathview::pathview(gene.data  = mutList,
                              pathway.id = "hsa04110",
                              species    = "hsa",
                              limit = list(gene=as.integer(max(abs(mutList)))))

hsa04010 = pathview::pathview(gene.data  = mutList,
                              pathway.id = "hsa04010",
                              species    = "hsa",
                              limit = list(gene=as.integer(max(abs(mutList)))))

hsa04115 = pathview::pathview(gene.data  = mutList,
                              pathway.id = "hsa04115",
                              species    = "hsa",
                              limit = list(gene=as.integer(max(abs(mutList)))))

hsa04151 = pathview::pathview(gene.data  = mutList,
                              pathway.id = "hsa04151",
                              species    = "hsa",
                              limit = list(gene=as.integer(max(abs(mutList)))))

hsa05224 = pathview::pathview(gene.data  = mutList,
                              pathway.id = "hsa05224",
                              species    = "hsa",
                              limit = list(gene=as.integer(max(abs(mutList)))))
```

A.8 Integration of all data and visualization

```
setwd("~/Desktop/TFM/")
workdir = getwd()
dataDir = file.path(workdir, "Data/")
robj_dir = file.path(workdir, "Robjects/")

library(maftools)

load(file = file.path(robj_dir, "common_patients.rda"))
load(file = file.path(robj_dir, "trans_results.rda"))
load(file = file.path(robj_dir, "meth_results_unique.rda"))
load(file = file.path(robj_dir, "cnv_results.rda"))
load(file = file.path(robj_dir, "mut_results.rda"))

trans_genes = dataDEGs$external_gene_name
meth_genes = unique(as.character(unique_df$Gene_Symbol))
cnv_genes = unique(AmpDel_genes$GeneSymbol)
mut_genes = unique(geneMat$Hugo_Symbol)

all_genes = unique(c(trans_genes, meth_genes, cnv_genes, mut_genes))

# Score computation

scores = data.frame(Genes = all_genes)
scores$Genes = as.character(scores$Genes)
scores$Trans_score = 0
scores$Meth_score = 0
```

```
scores$cnv_score = 0
scores$mut_score = 0

# Transcription

score = c()
name = c()
for (i in 1:nrow(scores)) {
  if (scores$Genes[i] %in% dataDEGs$external_gene_name) {
    score = c(score, subset(dataDEGs$logFC, dataDEGs$external_gene_name == scores$Genes[i]))
  }
  else {
    score = c(score, 0)
  }
  name = c(name, scores$Genes[i])
}
names(score) = name
scores$Trans_score = score

# Methylation

colnames(unique_df)[10] = "status"
unique_df$status = as.character(unique_df$status)
unique_df$status = replace(unique_df$status, unique_df$status == "Hypermethylated", "-1")
unique_df$status = replace(unique_df$status, unique_df$status == "Hypomethylated", "1")
unique_df$status = as.numeric(unique_df$status)

score = c()
name = c()
for (i in 1:nrow(scores)) {
  if (scores$Genes[i] %in% unique_df$Gene_Symbol) {
    score = c(score, subset(unique_df$status, unique_df$Gene_Symbol == scores$Genes[i]))
  }
  else {
    score = c(score, 0)
  }
  name = c(name, scores$Genes[i])
}
names(score) = name
scores$Meth_score = score

# Cnv

AmpDel_genes$Aberration = replace(AmpDel_genes$Aberration,
                            AmpDel_genes$Aberration == "Amp", "1")
AmpDel_genes$Aberration = replace(AmpDel_genes$Aberration,
                            AmpDel_genes$Aberration == "Del", "-1")
AmpDel_genes$Aberration = as.numeric(AmpDel_genes$Aberration)

# Get only homogeneus recurrent gene aberrations

final_cnv_df = matrix(ncol = ncol(AmpDel_genes), nrow = 0)
colnames(final_cnv_df) = colnames(AmpDel_genes)
final_cnv_df = as.data.frame(final_cnv_df)
for (i in 1:length(cnv_genes)) {
  int_df = subset(AmpDel_genes, AmpDel_genes$GeneSymbol == cnv_genes[i])
  if (length(unique(int_df$Aberration)) == 1) {
    final_cnv_df = rbind(final_cnv_df, int_df)
  }
```

```
}

final_cnv_df = final_cnv_df[!duplicated(final_cnv_df$GeneSymbol),]

score = c()
name = c()
for (i in 1:nrow(scores)) {
  if (scores$Genes[i] %in% final_cnv_df$GeneSymbol) {
    score = c(score, subset(final_cnv_df$Aberration, final_cnv_df$GeneSymbol == scores$Genes[
  }
  else {
    score = c(score, 0)
  }
  name = c(name, scores$Genes[i])
}
names(score) = name
scores$cnv_score = score

scores$cnv_score = score

# Mut

geneMat$Score = (geneMat$Score - min(geneMat$Score))/
                (max(geneMat$Score) - min(geneMat$Score))

score = c()
name = c()
for (i in 1:nrow(scores)) {
  if (scores$Genes[i] %in% geneMat$Hugo_Symbol) {
    score = c(score, subset(geneMat$Score, geneMat$Hugo_Symbol == scores$Genes[i]))
  }
  else {
    score = c(score, 0)
  }
  name = c(name, scores$Genes[i])
}
names(score) = name
scores$mut_score = score

## FINAL SCORE

scores$Trans_score = abs(scores$Trans_score)
scores$Trans_score = (scores$Trans_score - min(scores$Trans_score))/
                     (max(scores$Trans_score - min(scores$Trans_score)))
scores$Final = 0.3*scores$Trans_score + 0.3*abs(scores$Meth_score) +
               0.3*abs(scores$cnv_score) + 0.1*scores$mut_score

save(scores, file = file.path(robj_dir, "Integration_results.rda"))

# Pathview

eg = as.data.frame(bitr(scores$Genes, fromType = "SYMBOL", toType = "ENTREZID",
                        OrgDb = "org.Hs.eg.db"))
eg = eg[!duplicated(eg$SYMBOL),]

## Get results for annotated DMGs
Genes = scores[scores$Genes %in% eg$SYMBOL, ]
Genes = Genes[order(Genes$Genes, decreasing = F),]
```

```
eg = eg[order(eg$SYMBOL),]

Genes$ENTREZID = eg$ENTREZID

## Subset the data we need
Genes_sub = Genes[,6:7]
Genes_sub$Final_transf = abs(1/log10(Genes_sub$Final))

## Create a named vector containing the genes and their score
genelist = as.numeric(Genes_sub$Final_transf)
names(genelist) = Genes_sub$ENTREZID

## Get the maps

hsa04110 = pathview::pathview(gene.data  = genelist,
                              pathway.id = "hsa04110",
                              species    = "hsa",
                              limit = list(gene=as.integer(max(abs(genelist)))))

hsa04010 = pathview::pathview(gene.data  = genelist,
                              pathway.id = "hsa04010",
                              species    = "hsa",
                              limit = list(gene=as.integer(max(abs(genelist)))))

hsa04115 = pathview::pathview(gene.data  = genelist,
                              pathway.id = "hsa04115",
                              species    = "hsa",
                              limit = list(gene=as.integer(max(abs(genelist)))))

hsa04151 = pathview::pathview(gene.data  = genelist,
                              pathway.id = "hsa04151",
                              species    = "hsa",
                              limit = list(gene=as.integer(max(abs(genelist)))))

hsa05224 = pathview::pathview(gene.data  = genelist,
                              pathway.id = "hsa05224",
                              species    = "hsa",
                              limit = list(gene=as.integer(max(abs(genelist)))))
```

Appendix B

Additional figures

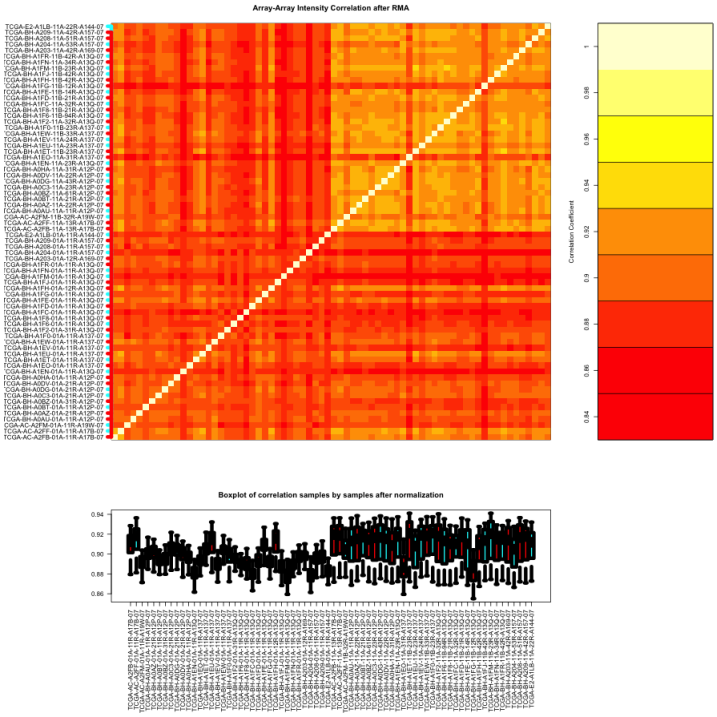


Figure B.1: Array-Array Intensity Correlation heatmap.

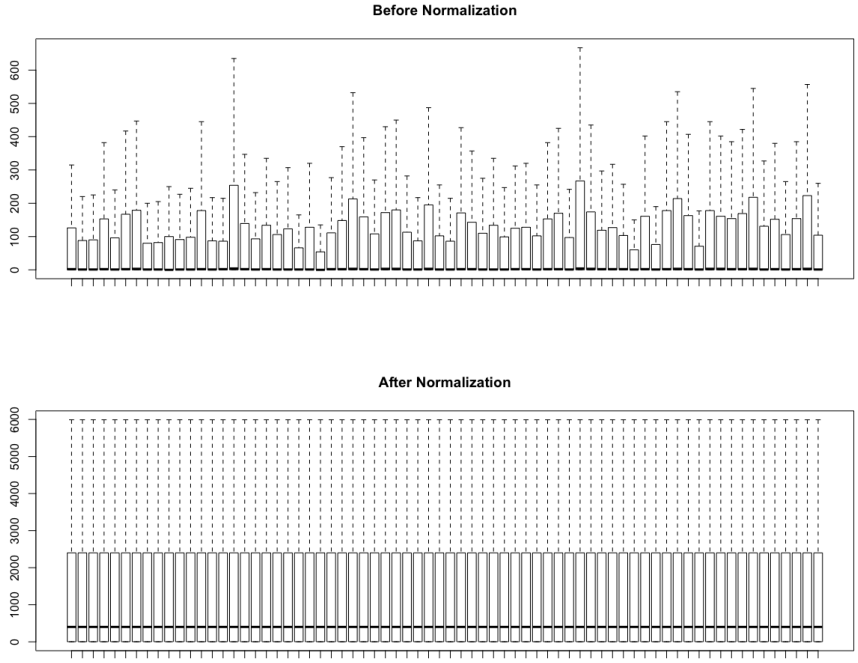


Figure B.2: Boxplots of the data before (top) and after (bottom) normalization was applied.

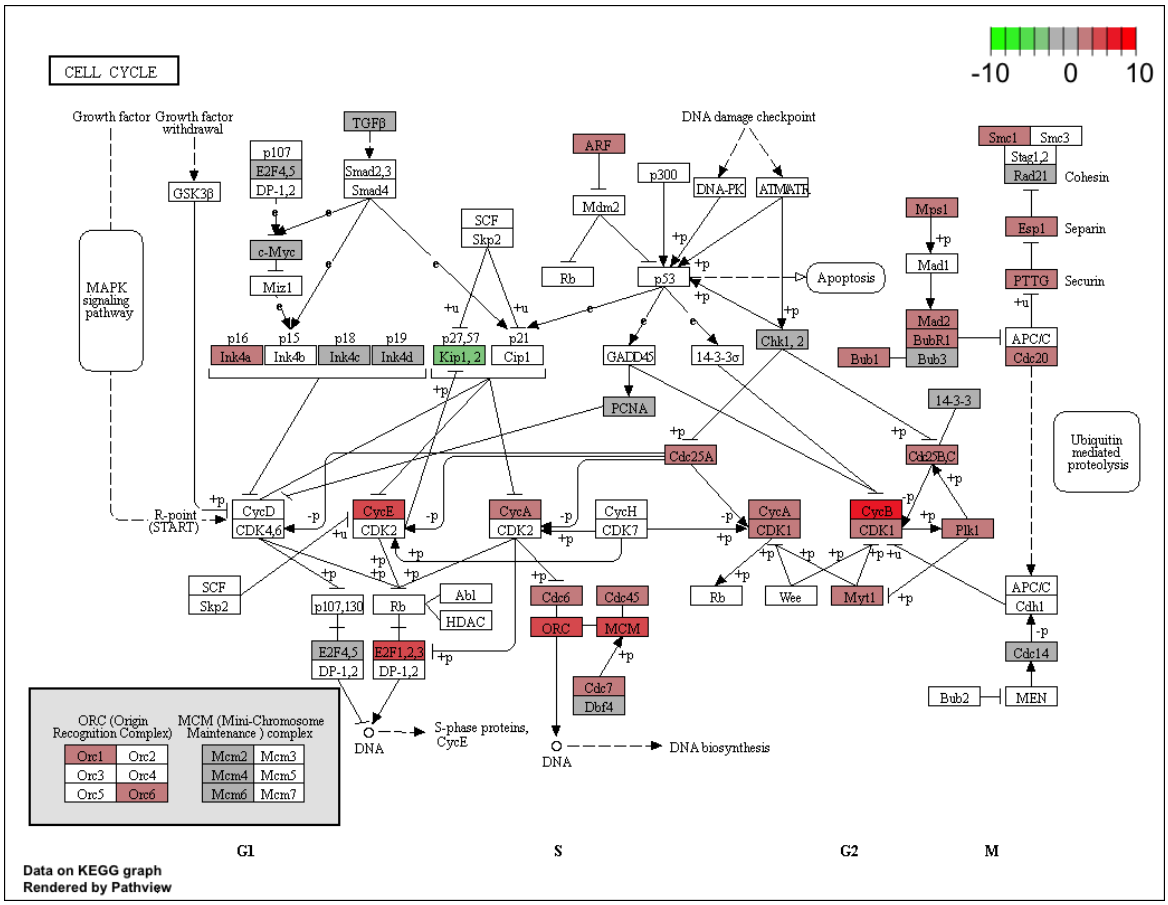


Figure B.3: The Cell Cycle KEGG pathway overlapped with the results of our differential expression analysis. Red nodes represent overexpressed genes and green nodes represent underexpressed genes. The intensity of the colour represents the logFC of the expression change.

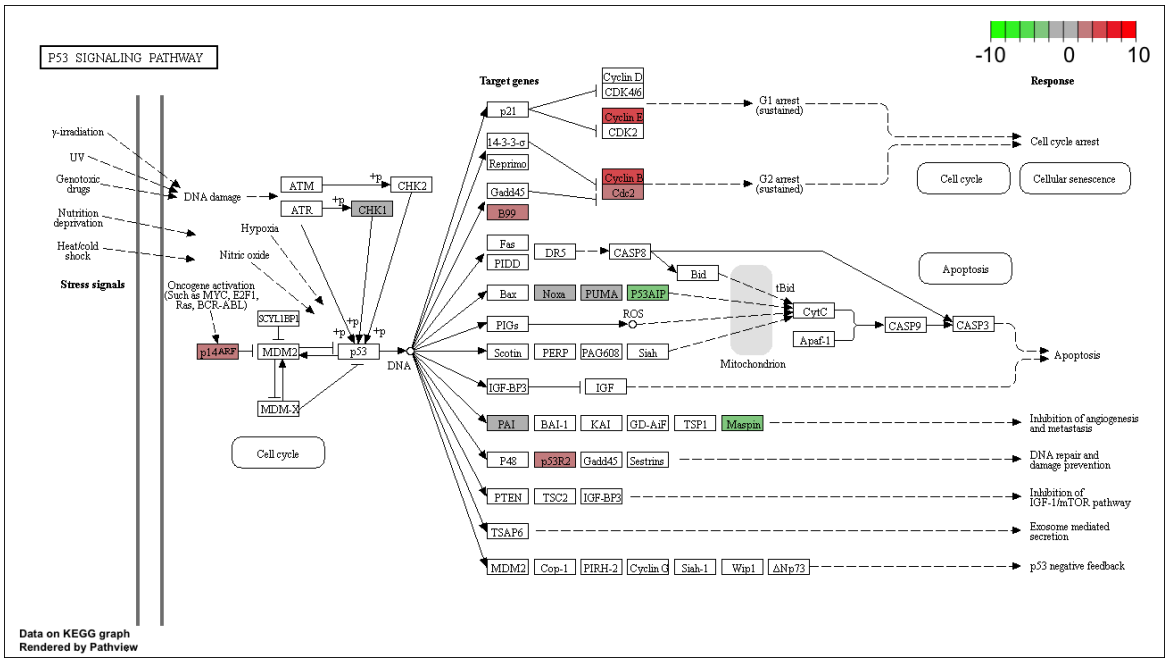


Figure B.4: The P53 KEGG pathway overlapped with the results of our differential expression analysis. Red nodes represent overexpressed genes and green nodes represent underexpressed genes. The intensity of the colour represents the logFC of the expression change.

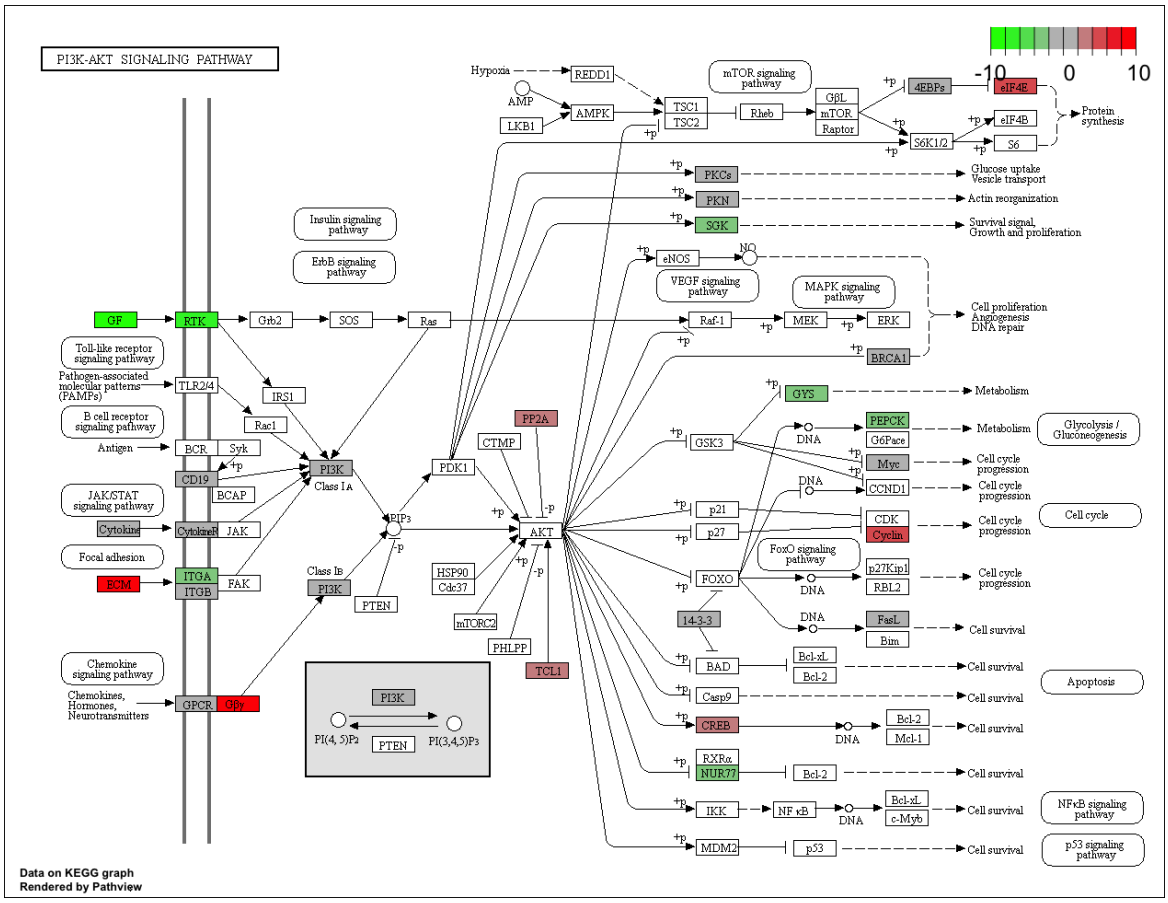


Figure B.5: The PI3K KEGG pathway overlapped with the results of our differential expression analysis. Red nodes represent overexpressed genes and green nodes represent underexpressed genes. The intensity of the colour represents the logFC of the expression change.

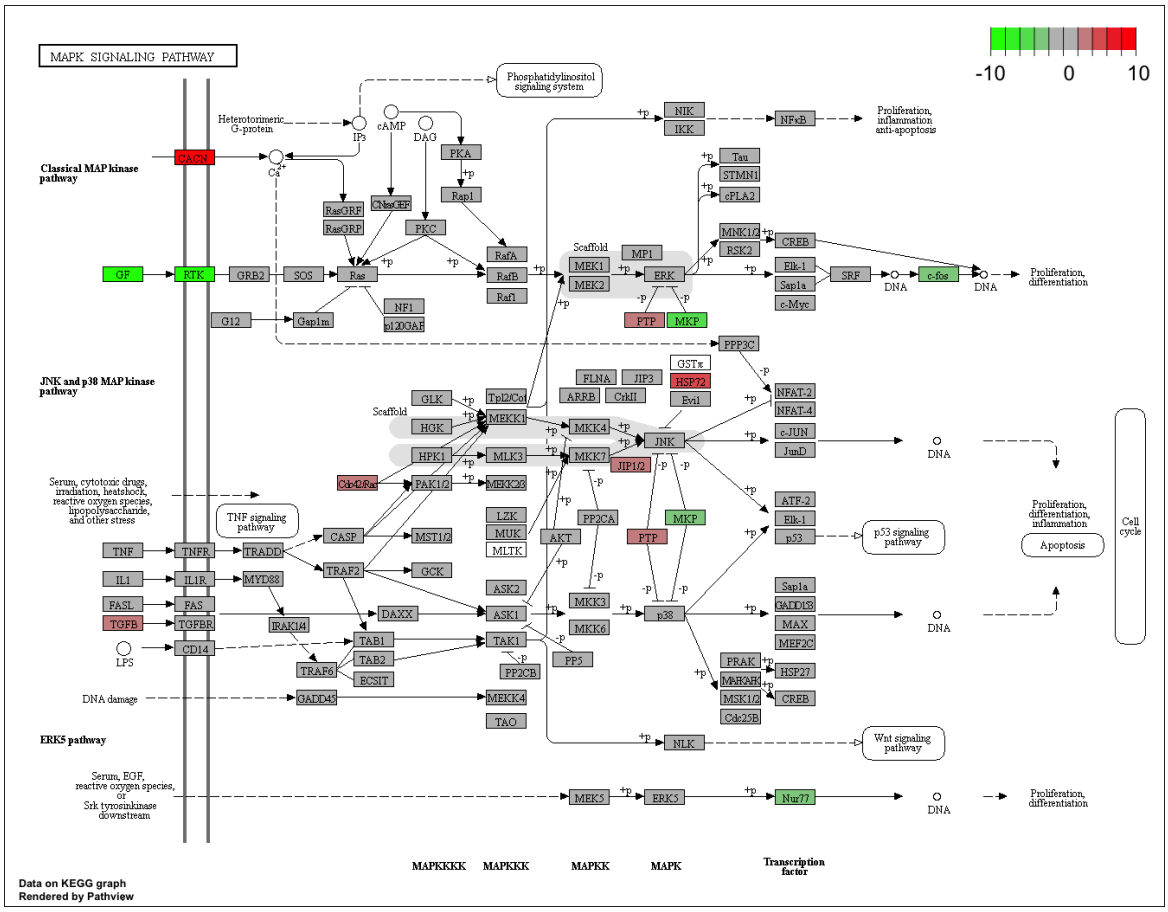


Figure B.6: The MAPK KEGG pathway overlapped with the results of our differential expression analysis. Red nodes represent overexpressed genes and green nodes represent underexpressed genes. The intensity of the colour represents the logFC of the expression change.

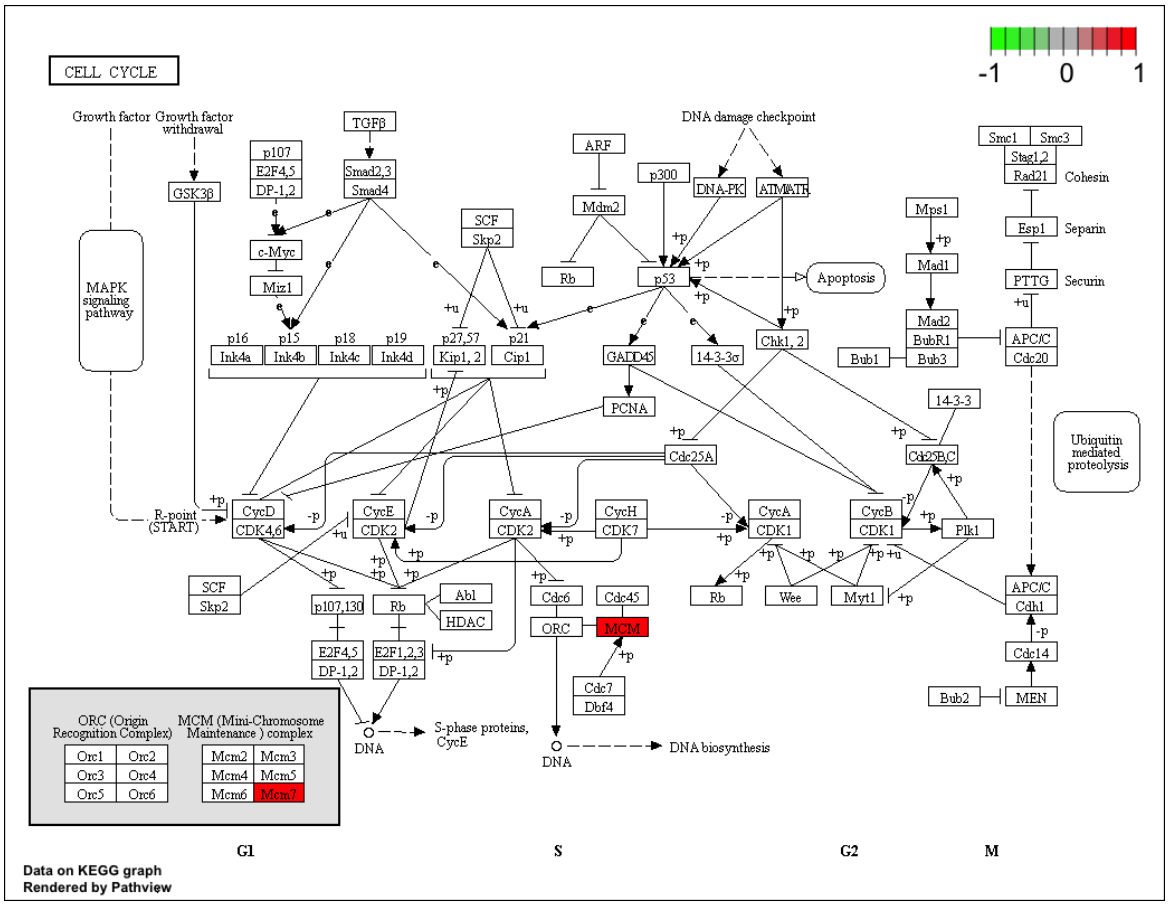


Figure B.7: Cell cycle pathway from KEGG overlayed with the results of the differential methylation analysis. Here, red nodes represent genes whose promoter seemed to be hypermethylated in the tumor samples, while green nodes represent genes whose promoter seemed to be hypomethylated in tumor samples.

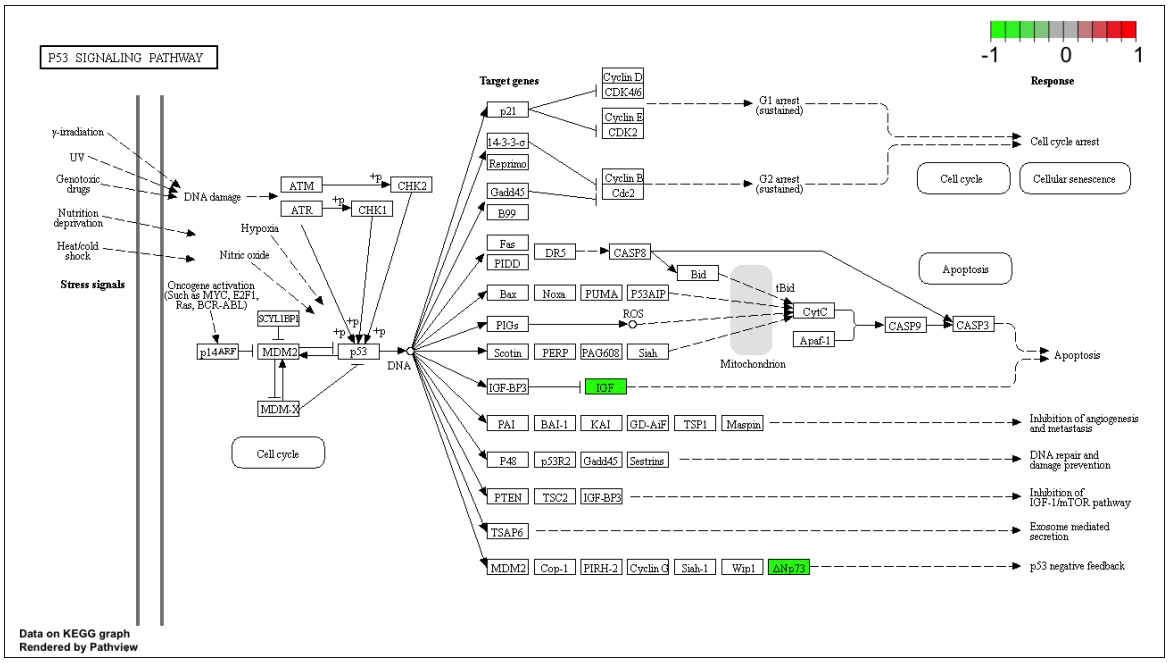


Figure B.8: P53 pathway from KEGG overlayed with the results of the differential methylation analysis. Here, red nodes represent genes whose promoter seemed to be hypermethylated in the tumor samples, while green nodes represent genes whose promoter seemed to be hypomethylated in tumor samples.

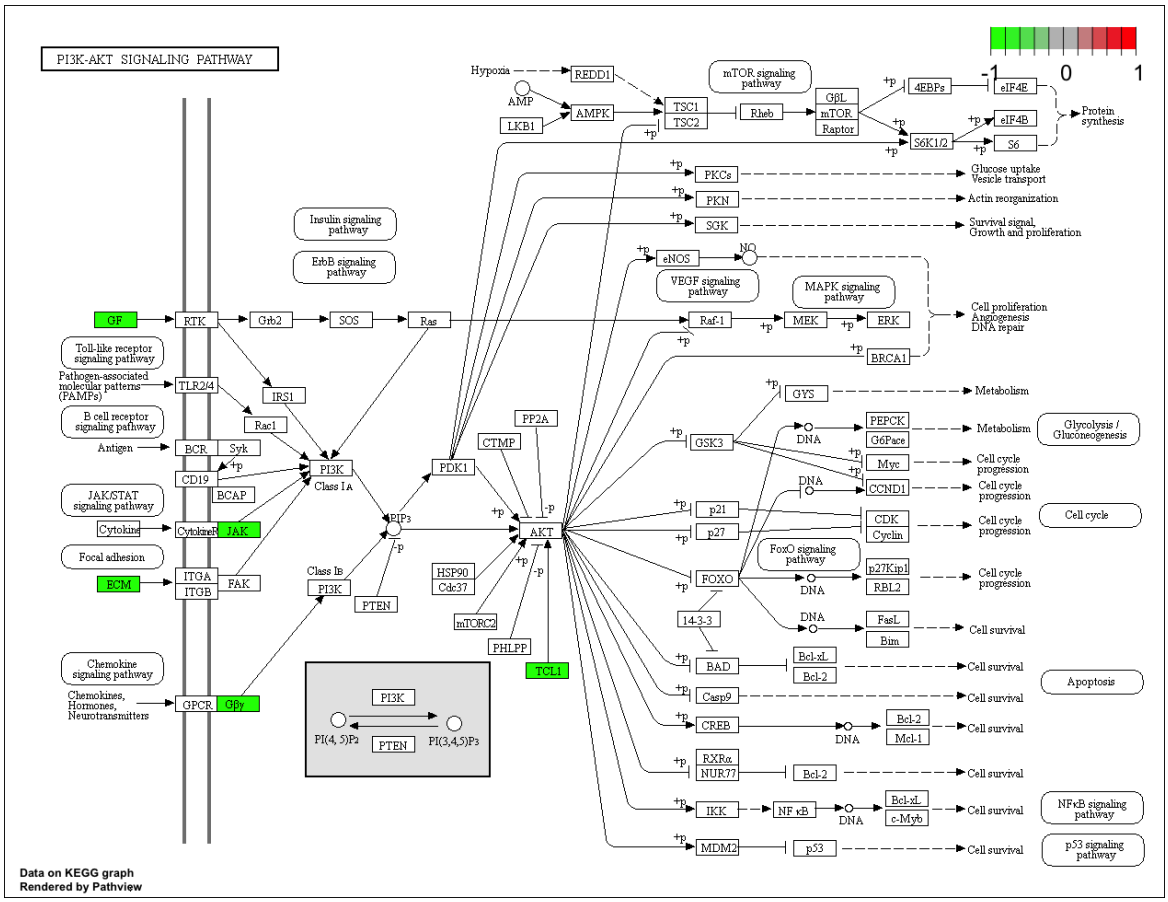


Figure B.9: PI3K pathway from KEGG overlayed with the results of the differential methylation analysis. Here, red nodes represent genes whose promoter seemed to be hypomethylated in the tumor samples, while green nodes represent genes whose promoter seemed to be hypermethylated in tumor samples.

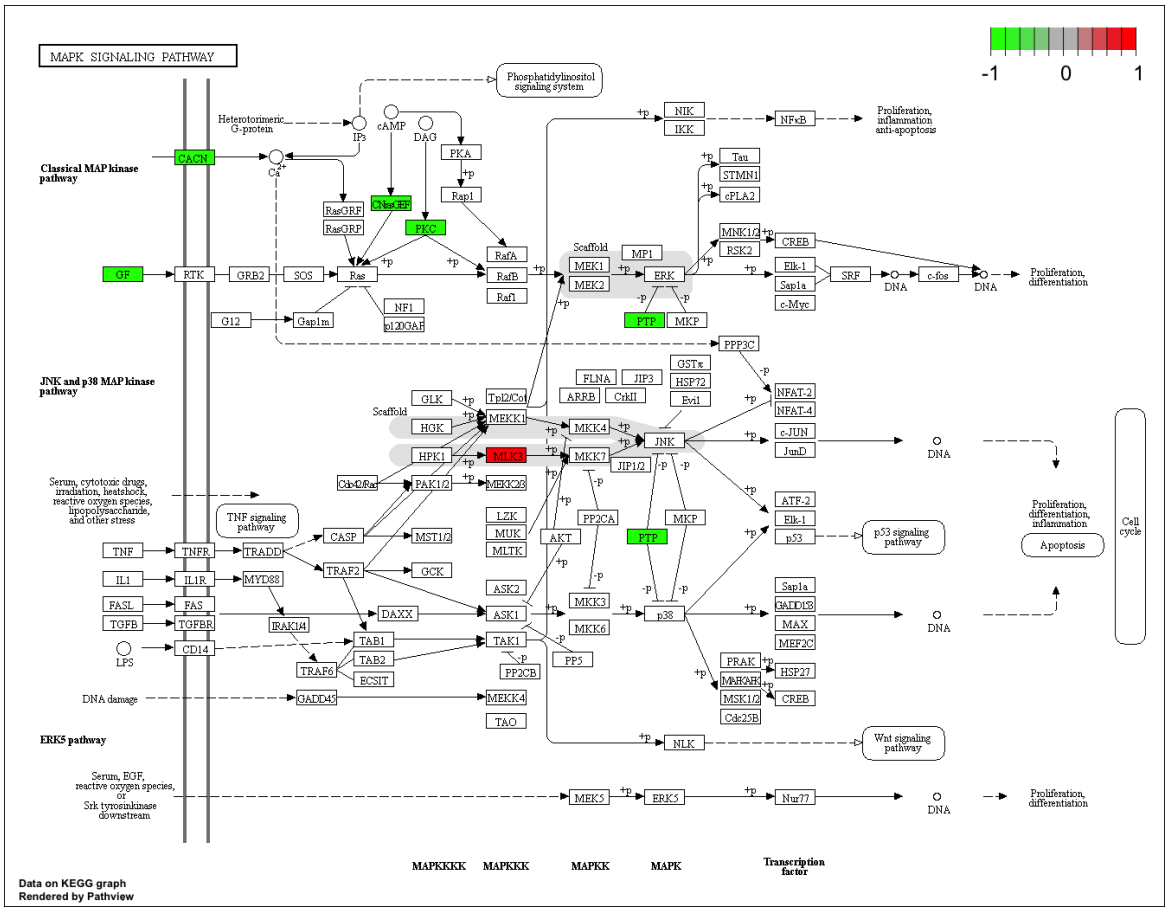


Figure B.10: MAPK pathway from KEGG overlayed with the results of the differential methylation analysis. Here, red nodes represent genes whose promoter seemed to be hypermethylated in the tumor samples, while green nodes represent genes whose promoter seemed to be hypomethylated in tumor samples.

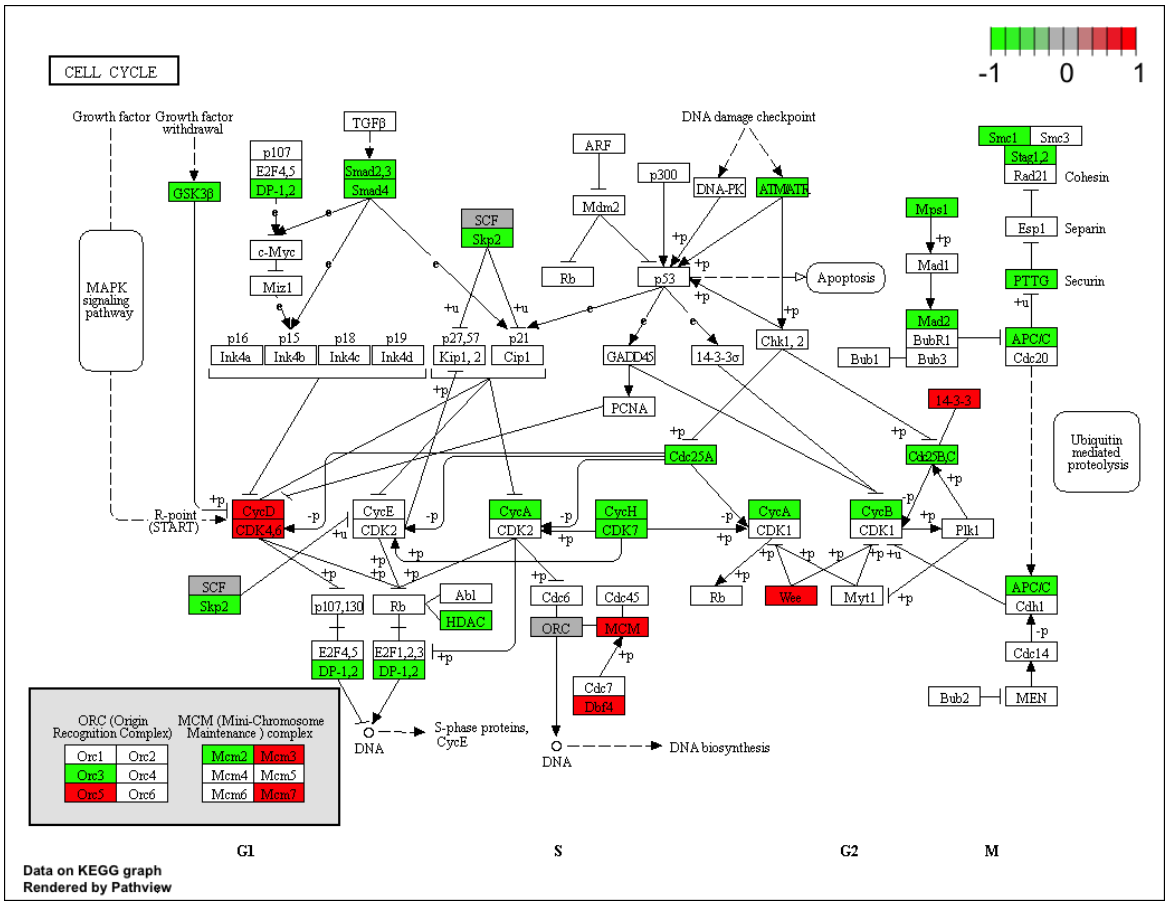


Figure B.11: Cell cycle pathway overlapped with the results of our recurrent aberration analysis. Red nodes represent genes that seemed to show recurrent amplifications in tumor, while green nodes represent genes that seemed to show recurrent deletions in the tumor samples.

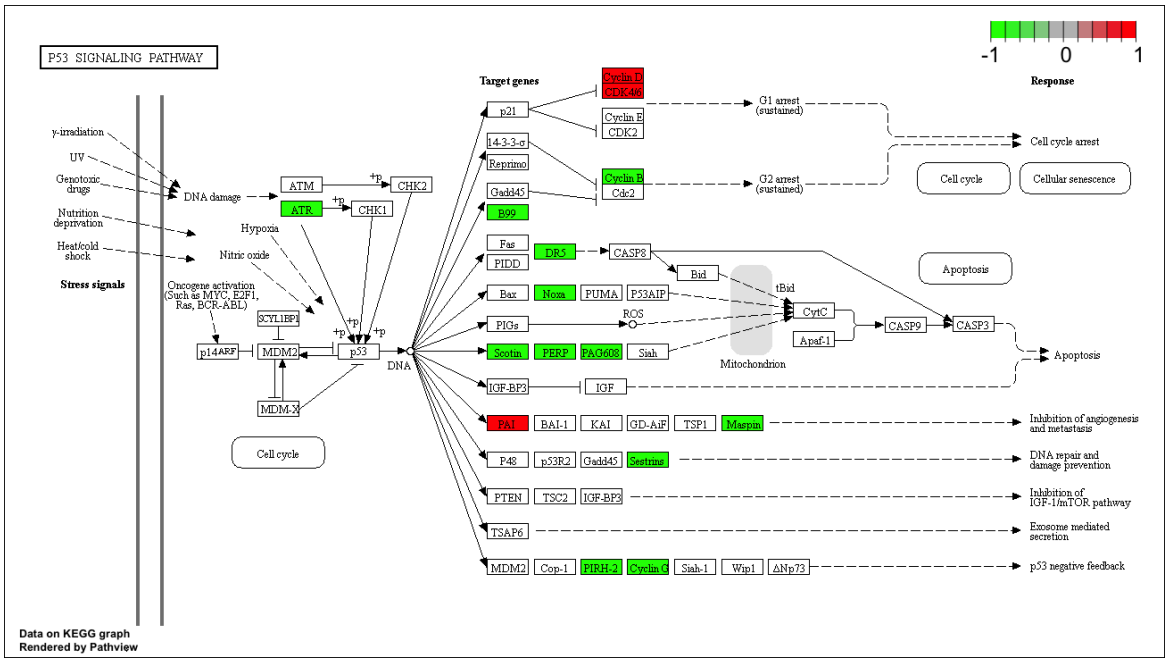


Figure B.12: P53 pathway overlapped with the results of our recurrent aberration analysis. Red nodes represent genes that seemed to show recurrent amplifications in tumor, while green nodes represent genes that seemed to show recurrent deletions in the tumor samples.

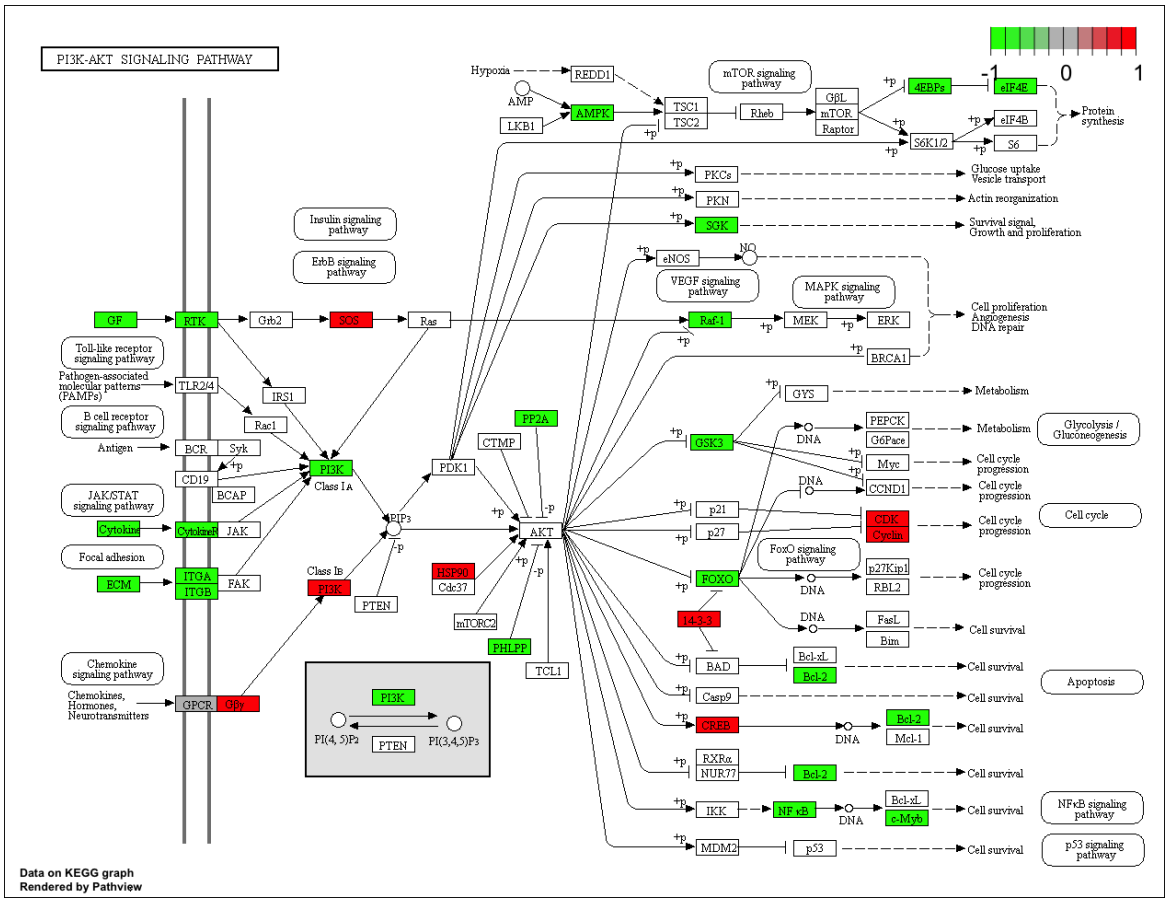


Figure B.13: PI3K pathway overlapped with the results of our recurrent aberration analysis. Red nodes represent genes that seemed to show recurrent amplifications in tumor, while green nodes represent genes that seemed to show recurrent deletions in the tumor samples.

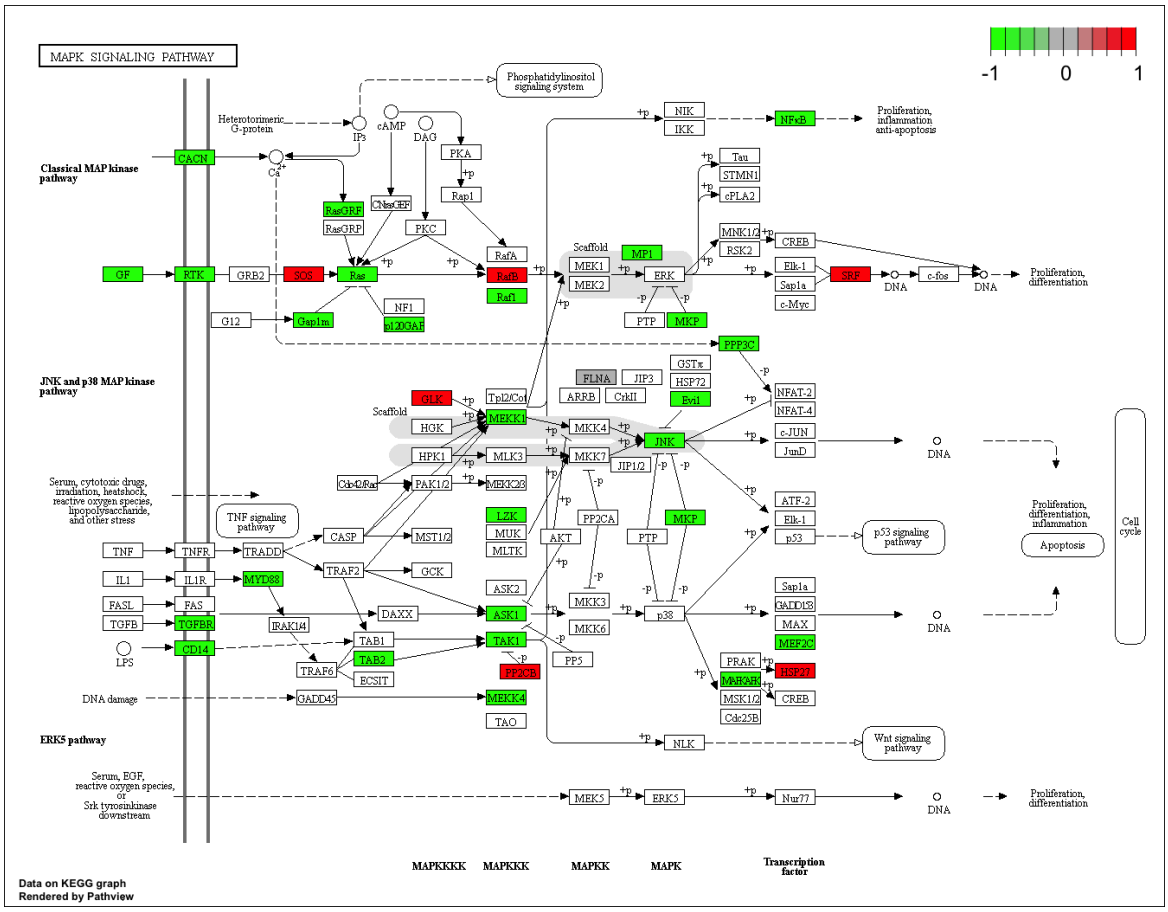


Figure B.14: MAPK pathway overlapped with the results of our recurrent aberration analysis. Red nodes represent genes that seemed to show recurrent amplifications in tumor, while green nodes represent genes that seemed to show recurrent deletions in the tumor samples.

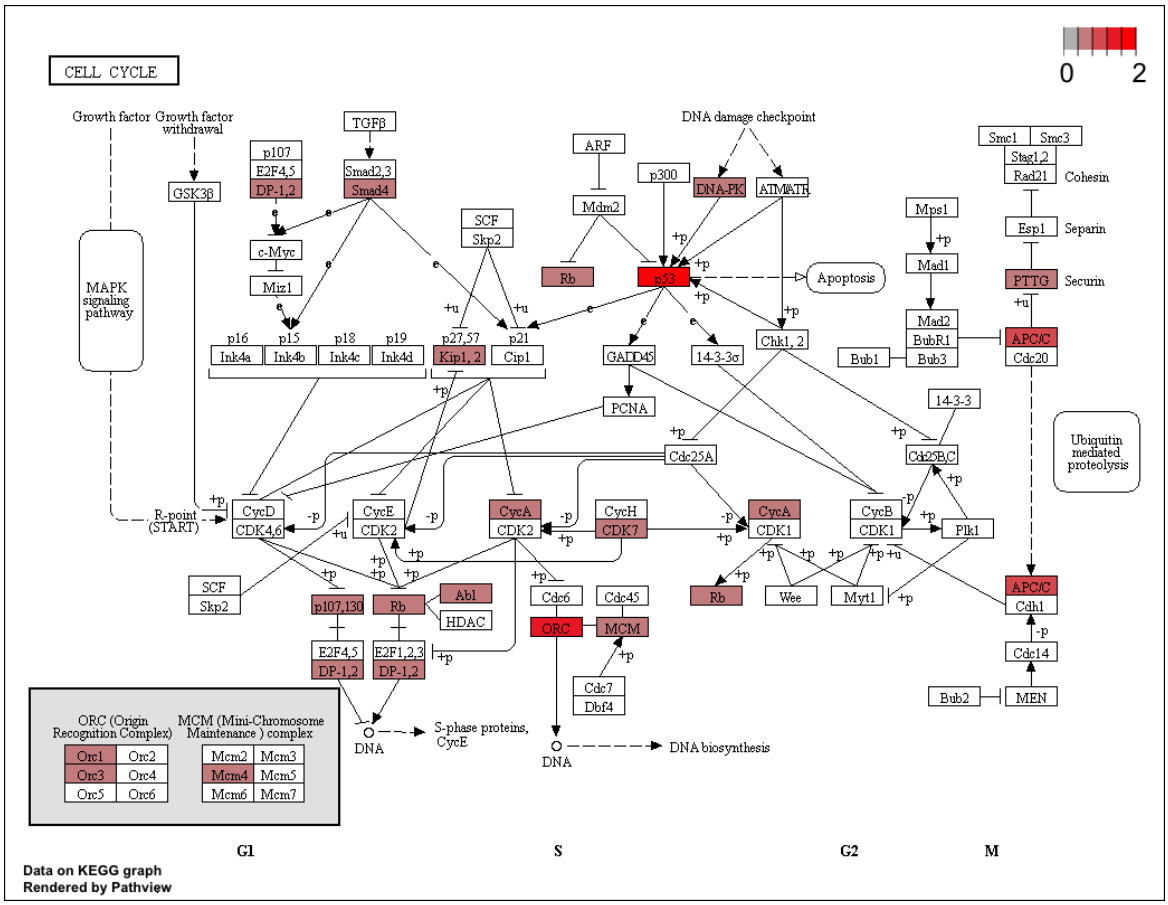


Figure B.15: Cell Cycle pathway for the DNA-seq data. Red nodes indicate genes that showed mutations in a higher proportion of samples, while gray nodes represent genes that showed mutations in a lower proportion of the samples.

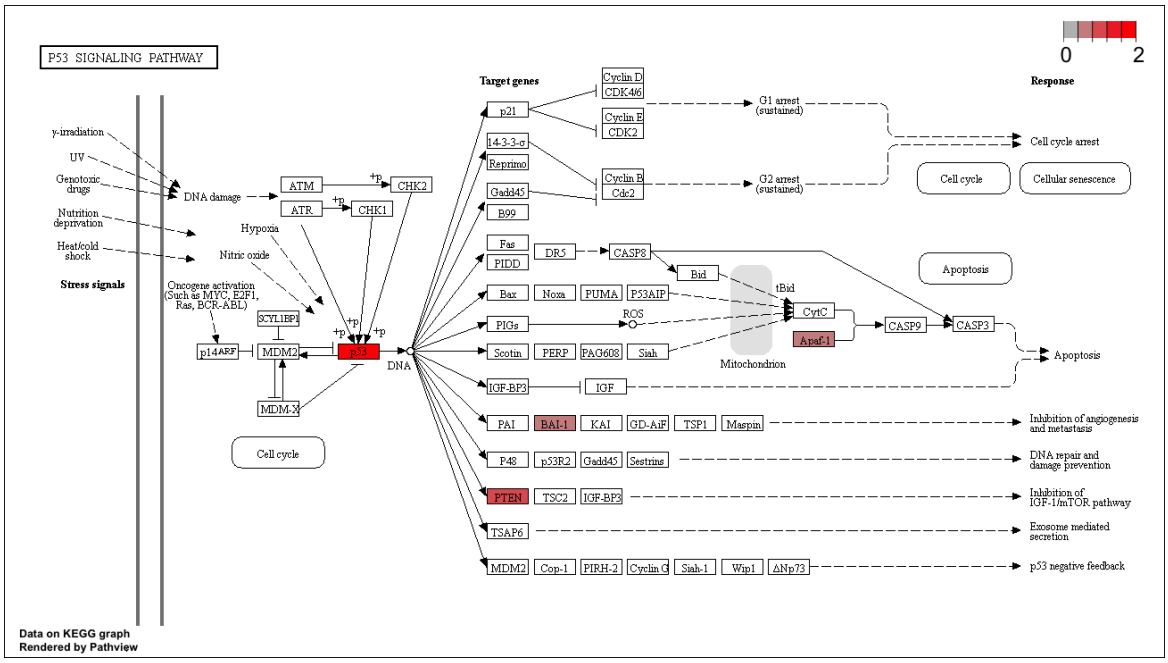


Figure B.16: P53 pathway for the DNA-seq data. Red nodes indicate genes that showed mutations in a higher proportion of samples, while gray nodes represent genes that showed mutations in a lower proportion of the samples.

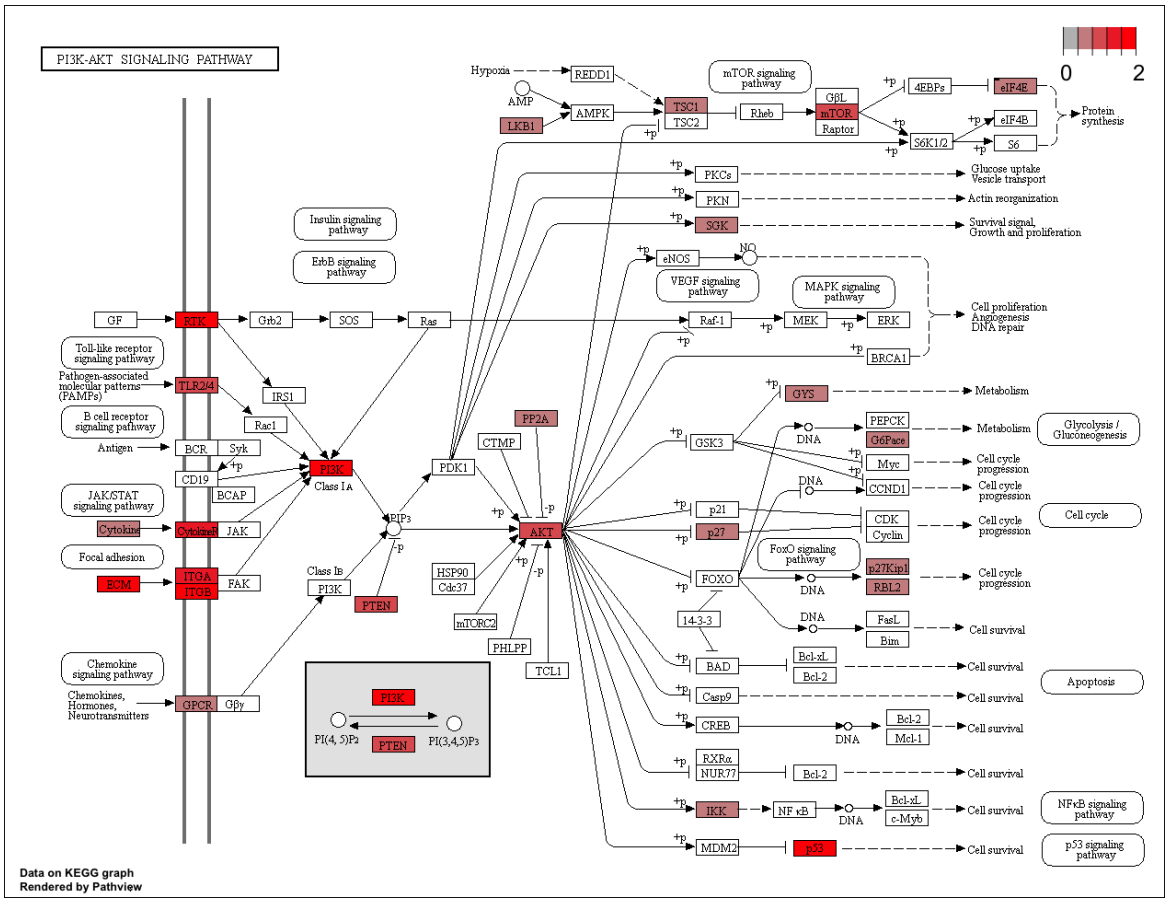


Figure B.17: PI3K pathway for the DNA-seq data. Red nodes indicate genes that showed mutations in a higher proportion of samples, while gray nodes represent genes that showed mutations in a lower proportion of the samples.

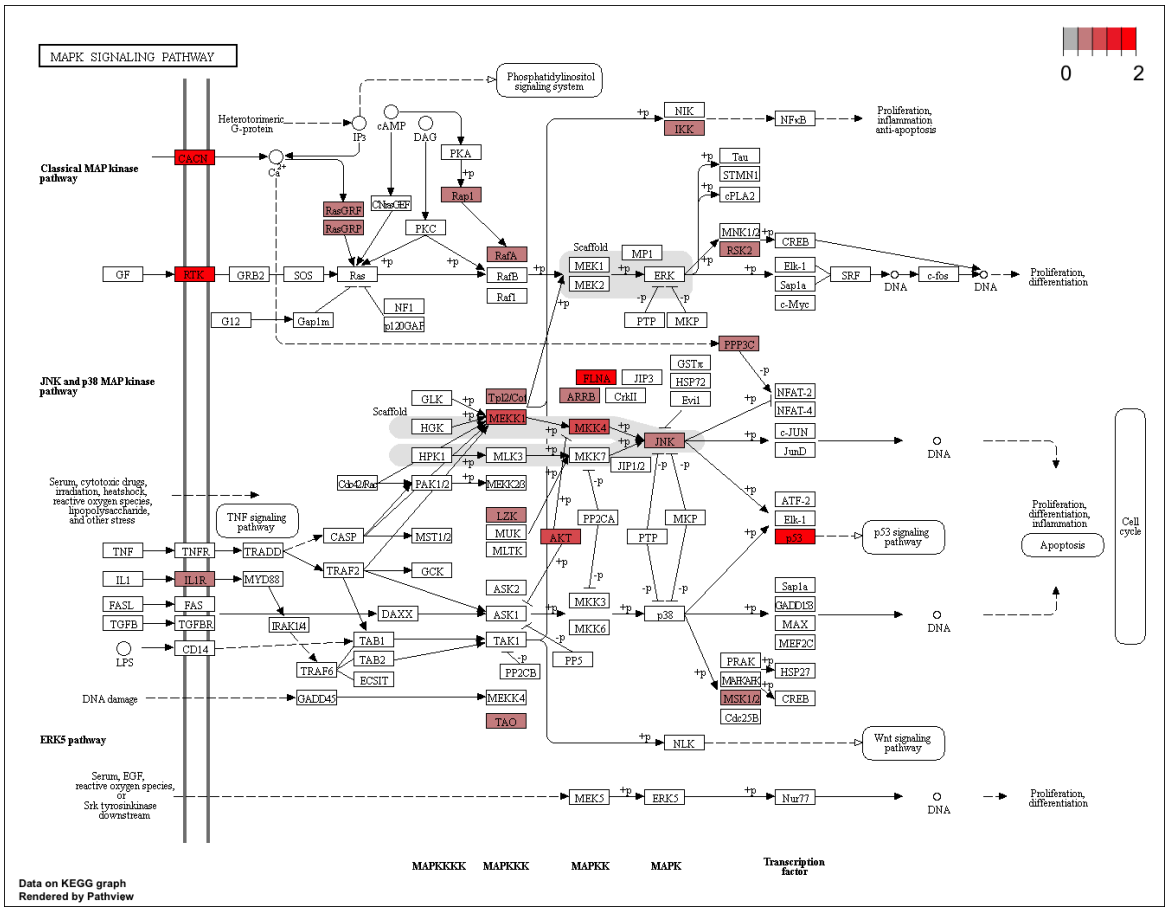


Figure B.18: MAPK pathway for the DNA-seq data. Red nodes indicate genes that showed mutations in a higher proportion of samples, while gray nodes represent genes that showed mutations in a lower proportion of the samples.